Unsupervised Online Concept Discovery in Structured Musical Streams

Duarte de Matos Soeiro Correia Teles

Thesis to obtain the Master of Science Degree in

Computer Science and Engineering

Supervisor: Doctor David Manuel Martins de Matos

Examination Committee

Chairperson: Doctor José Luís Brinquete Borbinha
Supervisor: Doctor David Manuel Martins de Matos
Member of the Committee: Doctor Bruno Emanuel da Graça Martins

January 2021

Acknowledgements

I would like to first thank my advisor, David, for all the support provided and for pushing me to do better. Not only in this work, he contributed deeply to the way i face challenges and more importantly, to my view of science. Throughout our many fruitful discussions, he always tried to bring the best in me...

To all the friends that supported me along the way, to my family, and everyone at HLT that always shared their knowledge and were always keen on explaining and giving more, i thank you.

Lisboa, January 22, 2021
Duarte Teles

For my family and friends. . .

Resumo

Uma peça narrativa pode vir acompanhada de música como forma de enfatizá-la. Neste trabalho, abordamos o problema de modelação da estrutura temática da música para conteúdo cinematográfico num cenário de *streaming*. Isto é concretizado por meio do mapeamento das relações entre conjuntos de personagens e locais em diferentes janelas temporais, elementos que afirmamos serem marcadores narrativos. Ao conectar situações semelhantes a partir da música que lhes está associada, relacionamos eventos narrativos por meio de sua similtude temática.

Apresentamos um método totalmente automático para gerar, a partir de um ou mais filmes e dos seus meta dados (guião e legendas), uma versão de qualidade do áudio que é reproduzida, associado a um conjunto de etiquetas. Estas podem ser usados de forma a mapear a narrativa do filme e, de forma mais geral, como uma verdade fundamental que pode ser aplicada a outros estudos. Generalizamos a verdade fundamental em termos da coocorrência de etiquetas. Isto permite-nos ter uma perspectiva topológica das diferentes diretrizes narrativas que ocorrem ao longo do filme.

Grupos de eventos semelhantes atuam como um ponto de ancoragem ao qual associamos a música que é tocada. Por ter uma etiqueta singular para descrever grupos de características musicais, podemos construir associações entre estes grupos e dar-lhes nomes. Usamos estes grupos para construir um mapa global de relações entre eventos cinematográficos semelhantes, dadas as suas características musicais compartilhadas.

Abstract

A narrative piece can be accompanied by music as a way of emphasizing it. In this work, we approach the problem of modelling the thematic structure of music for film content in a streaming scenario. This is achieved through the mapping of relationships between sets of characters and locations in different time windows, elements that we claim to be narrative markers. By connecting similar situations based on the music that is associated with them, we relate narrative events though their thematic similarity.

We introduce a fully automatic method to generate, from one or more movies and their metadata material (script and subtitles), a quality version of the audio that is played together with a set of labels. These can be used to map the narrative of the movie and more generally, as a ground truth that can be applied to other studies, acting as the semantic to the material that they are associated with. We generalize the ground truth in terms of the co-occurrence of labels. This allows us to have a higher level overview of the different narrative guidelines that occur through the movie.

Clusters of similar events act as an anchor point in to which we associate the music that is played. By having a hard label to describe groups of musical features, we can build associations between these groups and give them names. We use these groups to build a global map of relationships between similar movie events, given their shared musical characteristics.

Palavras Chave
Keywords

Palavras Chave

Extração de Relações

Dataset de Música para Filmes

Agrupamento não Supervisionado

Aprendizagem Incremental

Musicologia Computacional

Keywords

Relationship Extraction

Film Music Dataset

Unsupervised Clustering

Online learning

Computational Musicology

Contents

1 Introduction **1**

1.1 Objectives . . . 1

1.2 Contributions . . . 2

1.3 Musical Meaning in Film Scores . . . 3

1.3.1 Leitmotif . . . 3

1.3.2 Music in Film Scores . . . 4

1.3.3 Diegetic Music . . . 5

1.4 Document structure . . . 5

2 Background and Related Work **7**

2.1 Audio Pre-processing . . . 7

2.1.1 Feature Extraction . . . 8

2.1.1.1 Mel-frequency cepstrum (MFC) . . . 8

2.1.1.2 Chromagram (Chroma) . . . 9

2.1.1.3 Constant-Q Transform (CQT) . . . 9

2.1.1.4 I-vectors . . . 10

2.1.1.5 Chords . . . 11

2.1.1.6 Pitch contours . . . 11

2.1.1.7 Sequence Modelling . . . 12

2.1.2 Segmentation . . . 14

2.2 Methods for Semantic Extraction . . . 19
2.3 Density Estimation . . . 25
2.4 Dirichlet Process . . . 28
2.5 Language-based Methods . . . 34
2.6 Summary . . . 35

3 Dataset Preparation **37**

3.1 Dataset . . . 37
3.2 Audio Preparation . . . 39
3.3 Ground Truth and Metadata . . . 40
3.4 Clustering Labels . . . 45
3.5 Summary . . . 52

4 Experimental Setup **53**

4.1 Audio Setup . . . 54
4.2 Feature Extraction . . . 55
4.3 Computing Relationships . . . 56
4.4 Assessment . . . 56
4.5 Summary . . . 58

5 Experimental Results and Discussion **59**

5.1 Implementation . . . 59
5.2 Results . . . 60
5.3 Discussion . . . 64
5.4 Summary . . . 67

6 Conclusions and Future Work **69**

6.1 Conclusion 69

6.2 Future Work 71

A Cluster decoding **81**

List of Figures

2.1 Overview of the proposed recurrent autoencoder. Adapted from Amiriparian, Freitag, Cummins, and Schuller (2017). 13

2.2 Boundary algorithm performance given the type of feature, adapted from (Nieto & Bello, 2016). 16

2.3 Sampling distribution for i_n and i_p, for a given choice of i_a. δ_p and δ_n are constants, (McCallum, 2019). 18

2.4 Overview of the KDE model. Adapted from (Kristan, Leonardis, & Skočaj, 2011) 27

2.5 Graphical model overview of different constructions of the Dirichlet process. . . 29

3.1 Example of alignment between subtitles and script. 41

3.2 Map of the fictional world of the Lord of the Rings. From Tolkien (1991). 44

3.3 Crosses correspond to elements of the matrix C_d. Dots correspond to items of matrix F_d and are coloured given the cluster centroids they are closer to. 46

3.4 Projection of F_d items coloured by KNN clustering on C_d matrix.Crosses correspond to the centroids of the clusters obtained. 49

3.5 Projection of F_d items coloured by KDE clustering on matrix C_d. 49

3.6 Coverage of each cluster through the movies timeline. Clusters were generated using KNN method. 50

3.7 Coverage of each cluster through the movies timeline. Clusters were generated using KDE method. 51

4.1 Diagram of the pipeline of our work. 53

4.2 Top relationships computed between cluster 3 and all others. Thickness of edge represents more weight. 57

5.1 (a) Distance matrix computed using chroma features and with KNN clustering on the label space. (b) Chord diagram of top relationships between clusters. . . . 60

5.2 (a) Distance matrix computed using MFCC features and with KNN clustering on the label space. (b) Chord diagram of top relationships between clusters. . . . 61

5.3 (a) Distance matrix computed using Chroma+MFCC features and with KNN clustering on the label space. (b) Chord diagram of top relationships between clusters. 61

5.4 (a) Distance matrix computed using Chroma+MFCC features and with KDE clustering on the label space. (b) Chord diagram of top relationships between clusters. 62

5.5 (a) Distance matrix computed using Chroma+MFCC features and with KDE clustering on the label space and KDE used model the underlying audio of each cluster. (b) Chord diagram of top relationships between clusters. 63

5.6 (a) Distance matrix computed using MFCC features and with KDE clustering on the label space, and KDE used model the underlying audio of each cluster . (b) Chord diagram of top relationships between clusters. 64

5.7 Content of label cluster 0. Labels are sorted by frequency. Longer bar indicates higher frequency. 65

List of Tables

2.1 Algorithms present in the MSAF tool and their roles, adapted from Nieto and Bello (2016). 16

3.1 List of locations selected as abstractions from finer grained ones. 43

3.2 Mixture weights per component and corresponding coverage of each cluster. . . 48

A.1 Cluster decoding for label clusters computed with KNN method. 83

A.2 Cluster decoding for label clusters computed with KDE method. 86

1 Introduction

Music that accompanies a narrative, such as the case of music in films or operas, carries a thematic structure that helps guide the visual content it is associated with. The music chosen for a given conceptual representation carries specific features that imply that there is a mapping between music and themes that underline a narrative. Building a structure that explains the different thematic occurrences throughout a score allows assigning the key narrative points to themes that most emphasise them. Music directors usually use their artistic sensibility to make or choose music in accordance with the dramatic guidelines and narrative of the visual piece, choices that often condition its success. The thematic structure created can vary in complexity depending on the choice of music. An example of this can be seen in classical music: it can be characterized by a broad spectrum of composition, not marked by a single beat that follows the song or by a fixed mode, in contrast with pop music, that is characterized by tonality and repetitions branching from popular music, that make it much easier to process computationally. These aspects make classical music more challenging than other musical genres and so the techniques that work for processing pop music may have worse performance when attempting to segment or to find relevant transitions (Chai & Vercoe, 2005).

Regarding the musical stream timeline, different sound concepts may emerge, depending on the director's choice, such as the introduction of a specific musical segment that is played every time a given character appears. Events like these motivate us to understand these patterns and to develop a method that is able to identify thematic concepts in a musical stream as well as to incorporate newly observed ones, producing a structure that explains these concepts and is able to relate similar patters across the timeline.

1.1 Objectives

We assume a context where there is no prior information regarding the number of themes in the music, choice that is motivated by the fact that thematic structure can vary greatly depend-

ing on the context of the narrative the music is following. Consequently, methods capable of dealing with an unknown number of concepts are required. The structural analysis is aimed at being done in an online setting with the goal of capturing the evolution and emergence of concepts in the music, and how these relate to the story that is being told through the main material. This aspect also allows to study: 1. how the observed concepts relate to previously observed ones; 2. to follow the themes that are recurrent during the movies; 3. where they are the most relevant; 4. to map the occurrences of similar musical events. Furthermore, when using online methods, as data arrives, the model should be updated in order to reflect changes made by new observations. This update, in cases where the volume of information received in stream is too large to be kept in memory, is required to be done on a representation of the data observed so far in the stream.

To achieve our goal, we aim at using density based methods as well as non-parametric statistical ones. These prove more complex than traditional segmentation models and will be used with the expectation of capturing patterns in the stream that other methods may not be able to model.

1.2 Contributions

The main contributions of this work include:

- Integration of multiple tools for processing temporal data, including script and subtitle alignment tool, script information extractor and audio alignment tool.
- Introduction of a fully automatic method to generate, from one or more movies, a quality version of the audio that is played, given the corresponding soundtrack.
- Group movie events based on the co-occurrence of narrative information, specifically characters and locations.
- Build a map of relationships between sets of similar movie events, based on their musical similarity.

1.3 Musical Meaning in Film Scores

Films, among other forms of content, because they encase a fictional world where a narrative unveils, can have the dramatization of the its story complemented by music, as a way of elevating the narrative that it is being told. Some soundtracks are produced to create an atmosphere so the viewer can be immersed in the world of the movie. It has qualities especially well-suited to contribute to a films' narrative, as mentioned by Gorbman (1987), where "malleability, spatial, rhythmic and temporal values bond shot to shot, the narrative event to meaning, spectator to narrative and spectator to audience".

In the particular case of music composed to serve visual content, musical meaning becomes attached to the visual content and said meaning is retained by the observer when it is listened to in a detached manner from its original format. This detached semantic allows a piece of music, when paired with a piece of visual content such as a trailer or shot of a movie, different from what it was originally produced for, to deliver a similar semantic than when paired with its original counterpart. This property is used throughout many film instalments, whenever the main character, object or location, among others, are introduced in a scene or play a major role in it. There is an association between a given element in the film narrative and a corresponding track, that creates an expectation that that element will take some type of part in the narrative, whenever the music associated to that element is played.

1.3.1 Leitmotif

A motif corresponds to a recurring thematic event and in the musical domain, is called leitmotif. It is defined as a musical theme that lets the audience identify distinct musical material and relate it to a specific element in the story as a character or an object. It is the use of recurring and interrelated melodies that leads them to operate within a dramatic work according to an internal system of meanings linked to a narrative material. It can strengthen the connection to the narrative therefore making the content more appealing by carrying a stronger message.

The meaning can be the same or change throughout the play to reflect development since it not limited to an initial presence. Repetition and variation of melodic material help a leitmotif score achieve coherency (Bernanke, 2008).

1.3.2 Music in Film Scores

Films communicate (potentially) through a conjunction of visual and auditory signals. The music embedded in the auditory signals brings a broader dramatization of the events happening on screen. The narrative structure of a film is therefore deeply interconnected with the music that complements it, thus making the analysis of the narrative aspects of a movie relevant to understand how can music better emphasize these narrative aspects.

Seymour (1978) defines key concepts for the structure of a narrative. The author's concepts of kernel and satellite play a foundational role in the structure of narrative. Simply stated, a kernel is an event that captures a key point of the cause-and-effect structure of the story being told and a satellite is a non-key event that is a working out of a kernel. When applied to an entire film, they can produce a set of cause-and-effect relationships that clarify the structural role of all the events in that film.

Music can also locate a film's setting geographically. Popular melodies and folk tunes can conjure certain locations. Other musical devices can have a similar function. Depending on instrumentation, a pentatonic scale can suggest either the Oriental or the Native American for example, giving the film a tool to more easily change the narrative focus to a specific culture (Marks, 1979).

Films often have different themes for different entities, locations, or objects. These may be played in different variations, depending on the situation they represent, by changing the instrumentation used. This type of dynamics show the importance of the leitmotif and how it is impactful when transmitting ideas to the audience. There is an organization around abstract concepts that helps shape the different leitmotifs. Because these characterize specific aspects present in the movies, leitmotif occurrence is not uniform. The most popular themes in the film will have their respective leitmotifs being played proportionally to their impact on the narrative.

Marks (1979) also shows how film music is influenced by film genre. While genres have signature musical paradigms, these do not exist discretely, but in constant interaction with one another. Within a film, regardless of its dominant genre, narrative elements from other genres are at work. This implies that the music for a film, despite being mainly influenced by a narrative genre, carries influences from many others, as a way of complementing specific narrative

segments. There are also accounts where music does not fall into a predefined paradigm implying that this off-genre music works because of the familiarity with the conventions of the other generic paradigms shared by both the composer and audience.

Finer grained dependences on music can also be found when looking at film music, to induce specific emotions. An example of this is film's music dependence on and development of diatonic harmony. This concept is defined as chords or notes that relate to a certain key. For example, the note D is diatonic to the key of C because it can be found in the C major scale. Through the manipulation of this object, film music can create and deflate tension. The manipulation of volume or density of sound can also impact and dramatize a given scene.

1.3.3 Diegetic Music

Diegetic music corresponds to music that is heard or produced in the fictional world of the film. Although there is a possibility of extracting these from the stream and correlate them with the culture of the character or characters that produce it, the number of examples present may be insufficient for a data driven approach. Because our work deals mainly with film scores, mostly made up of orchestral music, these songs will be treated as the rest of the audio tracks and leitmotif extraction will be performed as well, along with the rest of the corpus.

1.4 Document structure

The rest of this document is structured in the following way: Chapter 2 overviews the related work that connect to our goal. This includes types of pre-processing that can be applied to the musical stream as well as state of the art models used for semantic extraction. It also covers background on the density-based and non-parametric methods. Chapter 3 presents all the steps taken to obtain and prepare the dataset. Finally, Chapter 4 presents the experimental setup of our work, followed by Chapters 5 and 6 where we show and discuss the results of our experiments and present some closing remarks, respectively.

2 Background and Related Work

In this chapter, we first present material in the literature related with the different challenges and goals of our work. We begin by exploring methods necessary to prepare the musical audio for further processing. These include methods for feature extraction targeted to our domain as well manners in which the audio can be portioned in similar segments. These aspects are covered in Section 2.1.

We then cover work in the literature that has dealt with some of the obstacles that we faced, including, novel class discovery, unsupervised clustering, and leitmotif classification, in Section 2.2. Finally, Sections 2.3, 2.4 and 2.5 cover the background models proposed as solutions to model estimation in our work that cover some of the work that utilizes these models.

2.1 Audio Pre-processing

The audio format we are working with is digital audio. A pre-processing step is required with the goal of preforming feature extraction. We require these features to be able to capture the harmonic, rhythmic, timbrel and sequence aspects of the music, as these help characterize a given musical theme. The features can be used as individual frames of the audio or further processing can be done in order to build audio segments or obtain structures such as chord-grams, both more complex objects that carry information of the sequence of frames.

In the case where we are dealing with music produced by an orchestra, the changes in timbre can be informative when attempting to extract semantically significant transitions. Key instruments in the orchestra are predominant in some of the leitmotifs. Because this feature allows us to distinguish instruments, similar patterns in the data are expected to be observed with the recurrence of a leitmotif, when using this feature. Moreover, different fictional cultures, in some cases due to the complexity of one's culture, may use different scales, mirroring what happens in real cultures, as it is the example of Asian and Western music. For a given

movie, music can be composed, for example, using the full chroma scale, on the opposite of music composed for another fictional cultures that may carry a different tonality of only 7 pitches, for example, (Rone, 2018). This suggests that this feature might be discriminate for modelling cultural aspects in music.

Choices following the literature baselines for either feature extraction or segmentation (Theodorou, Mporas, & Fakotakis, 2014) are not all available to us. Most methods in the literature take into account the full music, an assumption that can't be made in our context due to the online setting. Nonetheless, some state of the art methods, although prohibitive in our setting, are worth analysing as possible modifications can be done to bring these methods into our setting.

The rest of this section covers some of the methods in the literature for both feature extraction and segmentation.

2.1.1 Feature Extraction

If the musical stream is composed of mostly orchestral music or songs, a special attention to the features used is required since, as mentioned, extracting information from the audio is more challenging, compared to pop music.

In this section, we look at the lower level pre-processing steps that can extract information from the audio. There are multiple algorithms in the literature that attempt to discover structural relations in timbre, loudness, or harmony and combinations of these.

2.1.1.1 Mel-frequency cepstrum (MFC)

The Mel-frequency cepstrum models the subjective pitch and frequency content of audio signals. The Mel scale relates the perceived frequency of a tone to the actual measured frequency. It scales the frequency in order to match more closely what the human ear can hear. It is scaled following Eq. (1):

$$\text{Mel}(f) = 2585 \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.1}$$

where $\text{Mel}(f)$ is the logarithmic scale of the normal frequency scale f. The first constant (2585) is such that 1000Hz correspond to 1000 Mel, using a logarithmic scale with base 10. The sec-

ond constant (700) corresponds to the corner frequency where the scale changes from linear to logarithmic.

The MFCCs correspond to the coefficients that make up the MFC, and express the rate of change in each of the spectral bands. These are closely related to the timbre of the audio, making them relevant when trying to capture similarity between different instruments playing the same frequencies. They are computed by following steps:

1. Take the Fourier transform of a signal.
2. Map the powers of the spectrum obtained above onto the Mel scale, using triangular overlapping windows.
3. Take the logs of the powers at each of the Mel frequencies.
4. Take the discrete cosine transform of the list of Mel log powers, as if it were a signal.
5. The MFCCs are the amplitudes of the resulting spectrum.

2.1.1.2 Chromagram (Chroma)

Humans perceive musical pitches as similar if they differ by one or more octaves. A pitch class corresponds to the set of all pitches that are a whole number of octaves apart. Chromagrams combine the frequency components in the Short-Time Fourirer Transform belonging to the same pitch class and result in 12-dimensional representation corresponding to C, C#,D, D#, E, F, F#, G, G#, A, A#, B in music. These consist of the twelve pitch spelling attributes as used in Western music notation (A. K. Rasmussen, 2003).

Chroma features are used to aggregate, for a given local time window, all information that relates to a given chroma into a single coefficient. Shifting the time window across the music representation results in a sequence of chroma features, each expressing how the representation's pitch content within the time window is spread over the twelve chroma bands.

2.1.1.3 Constant-Q Transform (CQT)

Discrete Fourier Transform decomposes the audio signal into equally-spaced frequencies and provides corresponding intensities or amplitudes. The CQT is closely related and corresponds

to logarithmically spaced filters, where the width of each filter k is a multiple of the previous filter's width:

$$\delta f_k = 2^{1/n} \cdot \delta f_{k-1} \tag{2.2}$$

where δf_k is the bandwidth of the k_{th} filter and n is the number of filters per octave. The logarithmic spacing of bins of CQT is well-suited for musical data, since it provides a higher frequency resolution for low frequencies using many bins while rejecting higher frequencies using less bins.

2.1.1.4 I-vectors

The i-vector technique from Dehak, Kenny, Dehak, Dumouchel, and Ouellet (2010) is widely used in speaker recognition and verification. Assuming the audio space can be described by a Gaussian Mixture Model (GMM) with C components, the authors define a super-vector m of the C components that corresponds to the concatenated mean of every component. This GMM is called universal background model (UBM).

The goal of the i-vectors is to, given an audio segment, reduce the change of the posterior mean in the super-vector space when compared to the UBM. This can be modelled as:

$$\mu - m = Vy \tag{2.3}$$

where μ is the posterior mean of the UBM, V is the eigenmatrix and y is the i-vector. y will have a reduced dimensionality since it is assumed that an audio segment is only related to a subset of the components of the UBM, thus indicating that we can have audio features that have lower dimensionality and are approximately as representative as their full counterpart.

This technique allows us to perform a significant dimensionality reduction of our feature space, hence eliminating noise in the data, since the segments evaluated against the UBM will only produce variations in a subset of the model, indicating that the regions where that variance was found are the regions that better describe the evaluated segment. For the musical context, this method allows us to model musical sequence explicitly via a feature representation, information that is important for our task. Regarding the implementation of this technique in our work, the computation of the UBM must be taken into consideration. As we are in an online context, the UBM can be either computed with a different orchestral music dataset, or it can be

updated as new samples are observed.

2.1.1.5 Chords

A chord, in music, consists of any harmonic set of pitches consisting of multiple notes that are played simultaneously. A special case of chords are arpeggios and broken chords, in which the notes of the chord are played one after the other, rather than simultaneously. With automatic chord detection, the goal is to estimate chords from the observed notes in symbolic or acoustic form. Chord analysis is of interest because of its impact on harmonic content, a descriptive mid-level feature of (Western) music.

One method for chord extraction is proposed by Müller, Goto, and Schedl (2012). Firstly, the given audio piece is cut into frames and each frame is transformed into an appropriate feature vector. Most recognition systems rely on chroma-based audio features. Secondly, a matching method is used to map each chroma vector to a set of predefined chord models. One possible set consists on the twelve major and twelve minor triads.

2.1.1.6 Pitch contours

In cases where musical tracks contain singing (both diegetic and not), this characteristic may be an indicator of specific leitmotifs. Likewise, they can diverge greatly, since each of the cultures present in a fictional world possesses a background that alters the characteristics of the song and the themes in it. Like in the case of Chromagrams, where the scales used for the music of a given culture may change, so does singing possess clear structural differences. The amount of singing present in a film depends heavily on the type of soundtrack. For scored pieces, this number may be smaller than cases where pop music is used, for example. It is because of this aspect that using a feature that models singing aspects in particular, may improve the expressiveness of the data we will feed to the leaning model.

Albeit this feature may bring more expressiveness when the musical score relies heavily on singing pieces, such as the case of musicals, the number of tracks where music is present may be small compared to the overall number of tracks, which may generate noise when multiple features are used.

Panteli, Bittner, Bello, and Dixon (2017) focus on characterizing singing styles in folk and

traditional music by developing contour feature that model pith and melody. To achieve this goal, they use the extracted features to train a binary classifier to identify speech contours and then split them from non-vocal ones. These are then used to create a dictionary of singing style elements and each recording is summarized by a histogram with the proportion of each type of contour. K-means is then preformed in order to group similar recordings. Style connections are done a posteriori via metadata association.

Critically analysing these features, it is clear that global metrics can not be computed in our case, due to the online setting of the problem, resulting in features that only capture local behaviour such as local level changes of pitch via curve fitting, that summarize local direction of pitch. As an alternative, the model can be trained a priori with another dataset so that contours can be split on the fly, and used as features for our learning method, in an online setting.

2.1.1.7 Sequence Modelling

So far in this work, in regards to feature extraction, we have looked only at low level features and their aggregations, either by merging different descriptors or by building n-grams from small sequences of features. One key aspect that is important to be modelled is the melodic progression present in the audio, implying that we must be capable of modelling audio sequences. Another method for capturing such aspects is with the use of an autoencoder, so that the underlying structure of an audio segment, in the form of an embedding, can then further processed.

Amiriparian et al. (2017) follows this line of work of as they present a recurrent sequence to sequence model for learning unsupervised representations from audio. The authors first extract mel-spectrogram from the raw audio files and then proceed to train their proposed architecture with the extracted features, that are viewed as time-dependent sequences of frequency vectors. The learned representations of the spectrograms are then extracted for use as feature vectors for the corresponding instances. The task approached by the authors is acoustic event classification for the IEEE AASP challenge.

They use the extracted feature vectors to train a multi-layer-perceptron to preform the task. Moreover, as the dataset from the challenge contained audio sample recorded in stereo, the authors leveraged this aspect by repeating the aforementioned process on multiple channels

and performing fusion at the end, in the form of concatenation of the mean, left, right and difference features.

The architecture presented consists of an encoder RNN with N_l layers, each containing N_u Gated Recurrent Units (GRUs), introduced by Chung, Gulcehre, Cho, and Bengio (2014). Their final hidden states in each layer are concatenated into a one-dimensional vector. This vector can be viewed as a fixed-length representation of a variable-length input sequence. This vector is then passed through a fully connected layer with hyperbolic tangent activation. Finally another multilayered decoder RNN, similar to the one used for encoding, is used to reconstruct the original input sequence from the transformed representation. On the first time step, a zero input is fed to the decoder RNN. During subsequent time steps t, the expected decoder output at time $t-1$ is fed as input to the decoder RNN. The authors point out that this step accelerated model convergence.

The outputs of the decoder RNN are passed through a single linear projection layer with hyperbolic tangent activation at each time step, in order to map the decoder RNN output dimensionality to the target dimensionality. The weights of this output projection are shared across time steps. In order to introduce greater short-term dependencies between the encoder and the decoder, the decoder RNN reconstructs the reversed input sequence. The architecture just mentioned can be seen in Figure 2.1.

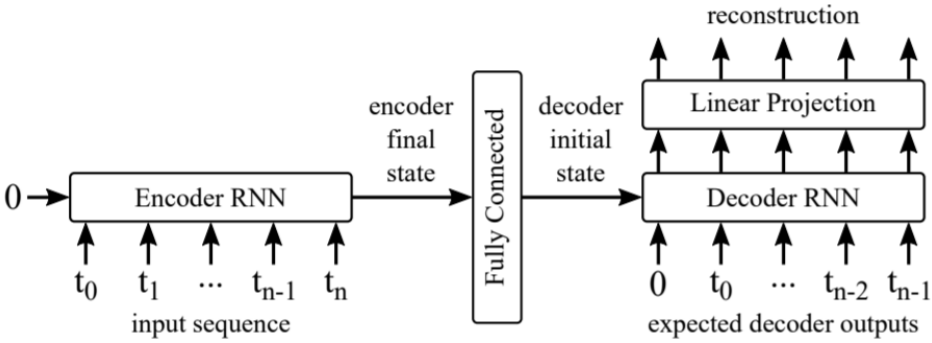


Figure 2.1: Overview of the proposed recurrent autoencoder. Adapted from Amiriparian et al. (2017).

Another mechanism that allows us to capture melody is proposed by Zalkow and Müller (2020). Here, the authors goal is preforming cross-version music retrieval, that aims at identifying all versions of a given piece of music. For this purpose, the authors propose two different techniques to approach this problem: one is based on classical principle component analysis,

and the other based on neural networks with triplet loss, both using a short audio fragment as query. Moreover, this work is focused on western classical music, which brings it closer to our work in terms of the dataset used.

Their approach is based on audio shingles, which are short sequences of feature vectors. These are generated from audio recordings, which are represented by longer feature sequences of variable length. In their work, the authors use chroma-based variant called CENS (chroma energy distribution normalized statistics), which are chroma features with post-processing applied: First, each chroma vector is l^1-normalized. Then, the resulting values of the chroma features are aggregated by mapping logarithmically spaced value ranges to integer values. Next, the chroma feature sequence is temporally smoothed. Finally, each chroma vector is then l^2-normalized. The authors argue that most important aspect of this post-processing is the temporal smoothing, because it makes the features more robust against tempo differences.

An audio query is then taken and split into multiple overlapping audio shingles. The retrieval approach is tested in three different manners. The first is a brute force approach where the shingle is compared directly with examples from an audio database, The second approach applies principle component analysis to the shingles and then makes the comparison, and finally, the third approach uses CNN based network with triplet loss for dimensionality reduction. This triplet loss used an anchor shingle, a positive example, corresponding to another interpretation of the same track and a negative example that is nor from the same piece or the same interpretation. In terms of results the authors show that both methods of dimensionality reduction benefit the task, with greater gains with the use of the CNN based network.

Despite showing results for their setup, this work cannot be fully adapted to our task. We would require multiple interpretations of the soundtrack of the movies in other to apply triplet-loss setup described, one of the key points from the work described.

2.1.2 Segmentation

All the features previously mentioned correspond to low level approaches to extract information from the audio signal. These can be built upon and form segments. We can define segments as a unit of a segment structure, bounded in time and comparable. In the same sense, a segment structure is a mental representation of music where segments are organized in either groups, chains or holarchies (Rodríguez López, 2016). Specifically, structural segmentation fo-

cuses on modelling temporal boundaries within an audio track that capture repetitions and similarities. This can be further divided into two sub-problems: boundary detection and structural grouping. The first identifies the beginning and end of a segment and the second labels the obtained segments based on acoustic similarity. For our task, we are interested on the first sub-problem, as mentioned, since decisions about segments similarity will be taken later in the learning process.

One could argue that structural segmentation could model the thematic structure of a musical stream. As we will see, the methods analysed here, although showing promising results in their specific settings, do not offer solutions for jointly dealing with an unknown number of concepts while at the same time being able to produce explanations of the observed ones.

Since we aim at building features that can encode information necessary to model leitmotifs, the use of segments may prove more descriptive than lower level features. Temporally close frames of a segment may provide more information that a sole frame in a bag-of-frames scenario, as the latter does not account for the sequence of notes in the music, information that, in our case, is relevant. As a result the set of features that can be extracted from structural segments provide further information.

Nieto and Bello (2016) propose MSAF, a framework for structural music segmentation containing the algorithms present in Table 2.1. They run a set of experiments where the algorithms shown are tested with multiple features. They evaluate them using tow different metrics: hit-rate, where estimated boundaries are considered hits if they fall within a time window from reference, and using Pairwise frame clustering, that compares each pair of frames by checking if they belong to the same cluster. In particular, for the case of boundary detection, the setup consists of using the annotated dataset "The Beatles TUT" (Mauch et al., 2009), following a constant sampling rate and hop size.

This set of experiments thus gives us an overview of which algorithms preform better with each set of features, and can give some insight for our experimental setups when using the more common methods in the literature. Their results can be seen in Figure 2.2. Some of the algorithms for boundary detection implemented in this framework, are unsupervised and can function in our setting. In particular, the structural features and the checker-board kernel methods are in these conditions, making Figure 2.2 an indicator how these methods may behave in different conditions.

Algorithms	Boundary	Grouping
2D-Fourier Magnitude Coeffs	No	Yes
Checkerboard Kernel	Yes	No
Constrained Cluster	Yes	Yes
Convex NMF	Yes	Yes
Laplacian Segmentation	Yes	Yes
Ordinal LDA	Yes	No
Shift Invariant PLCA	Yes	Yes
Structural Features	Yes	No

Table 2.1: Algorithms present in the MSAF tool and their roles, adapted from Nieto and Bello (2016).

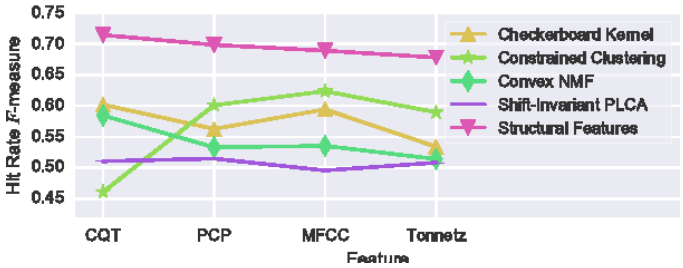


Figure 2.2: Boundary algorithm performance given the type of feature, adapted from (Nieto & Bello, 2016).

On the opposite of the majority of the methods exposed above, that require the full music in order to function, we require segmentation methods capable of dealing with an unknown number of segments (unsupervised) and that are able to function in the online setting, basing their decisions locally, given a time-window.

Krymova, Nagathil, Belomestny, and Martin (2017) present a procedure that detects changes in the eigenspace structure of the constant-Q spectral representation given the degree of explained variance. Their goal is to reconstruct the signal using a low rank approximation via principle component analysis (PCA), so that the obtained signal carries only relevant information. To achieve this, an intermediate pre-processing step is required where their proposed segmentation method is used.

They represent the CQT-based time frequency as $X \in \mathbb{C}^{N \times K}$ where N is the number of frames and K is the number of frequencies. This matrix is segmented into M non-overlapping blocks. By applying PCA to matrix X they obtain matrix U that is the projection of M onto orthogonal basis such that it represents a high amount of the total variance contained by the original matrix. U is explained by the first n dimensions of the transformed space that represent

most of said variance. With this, the dimensions on the projected space will capture most of the temporally correlated harmonics and n is chosen as the number of dimensions that can capture a large proportion of the total variance.

With this assumption, the authors break matrix U into two sub-blocks $U_1 \in \mathbb{R}^{B_m^1 \times K}$ and $U_2 \in \mathbb{R}^{B_m^2 \times K}$, where B_m is the number of frames in the matrix. They follow

$$R(\hat{k}, \mathbf{W}_1, \mathbf{U}_1) - R(\hat{k}, \mathbf{W}_1, \mathbf{U}_2) < \delta \tag{2.4}$$

where $\hat{k}$ is the number of principal components, W_1 is the projected matrix and δ is a constant. The function $R(\hat{k}, \mathbf{W}, \mathbf{U})$ corresponds to the variance ratio between $\mathbf{W}$ and $\mathbf{U}$. The authors argue that the value of δ is empirical but that higher values will lead to small number of components.

McCallum (2019), attempting to circumvent the problem of lack of labelled data for the segmentation task, proposes to solve this problem using unsupervised audio feature embeddings, the result of a method which aims at learning discriminative embedding features without human annotated labels. They use unsupervised training of Convolutional neural network (CNN) to obtain features for music segmentation that are more meaningful than lower level features. By exploiting the fact that musical segments form contiguous regions in a stream and that each musical label occurs in minor portion of a song, a sampling schema to create positive and negative examples to trains their model is used. This approach consists on sampling features that occur close together based on implicit time proximity information.

Given this premise, they use an anchor beat i_a, uniformly sampled from a set of beat indices in the song denoted by $\{0..L-1\}$ where L is the number of beats. A positive beat i_p index and a negative one, i_n, are then sampled from the distributions in Figure 2.3, so that a set of examples that belong to the same segment as the anchor beat is obtained, while negative examples will come from other segments.

A comparison between the log-amplitude of the 2D Fourier transform of the log-amplitude of a 8-beat-long CQT segments is considered for every beat index. This comparison is preformed to inform the sampling of positive and negative examples and decrease the number of false positives. An Euclidean distance is measured between these CQT segments, two regular beat intervals before and after i_a, so that the side with minimum distance to i_a is chosen to

sample i_p and the other one is used to sample i_n.

To test the produced embeddings, regarding the identification of the boundaries of the segments, a self-similarity matrix (SSM) is constructed. The corresponding SSM is computed as

$$S[i,j] = \| f(x_i[q,k]) - f(x_j[q,k]) \|_2^2 \tag{2.5}$$

where $x_i[q,k]$ and $x_j[q,k]$ correspond to beat synchronous CQT frames centred at beats i and j and f corresponds to the transformation function of the CNN. To detect the boundaries, a checker-board kernel is convolved along the diagonal of the SSM, producing a novelty function. This kernel can be written as the difference between a "coherence" and an "anti-coherence" kernel. The first kernel measures the self-similarity on either side of the centre point (the diagonal region) and will be high when each of the two regions is homogeneous. The second kernel measures the cross-similarity between the two regions around the diagonal and will be high when there is little difference across the centre point. The difference between the two values estimates the novelty of the feature sequence at the centre point. The novelty is high when the two regions are self-similar but different from each other.

Finally, boundaries are detected as peaks in the novelty function at time positions where the kernel meets a transition between two contrasting blocks.

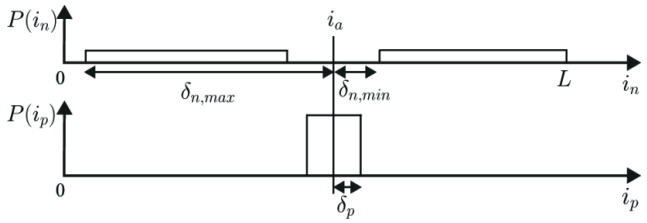


Figure 2.3: Sampling distribution for i_n and i_p, for a given choice of i_a. δ_p and δ_n are constants, (McCallum, 2019).

This method, despite the unsupervised approach, may have a poor behaviour in our scenario. Given that we are using classical music, there might be a poor estimation of the beat indexes. Furthermore, the training of the model would have to be done with the full discography or with a part of it so that we can achieve more representative segments which is not possible in an online setting. A plausible modification is to use apply the model to local batches of the musical stream.

Another possible approach is to perform blind segmentation where we produce segments

of fixed size. Huang, Cheng, Li, Hautamäki, and Lee (2013) propose a method to label multiple acoustic events contained in equal-length segments. The key point is the use of i-vectors described earlier as features that model those segments. Since lower variance in some dimensions of the i-vector indicates a close similarity to a subset of distributions in the UBM, distinct events can be captured with this approach.

2.2 Methods for Semantic Extraction

Having proposed a set of representations for our corpus, either based on frames, segments, or chords, these can now be applied to capture meaningful structure using a learning model.

The number of leitmotifs in the stream can grow indefinitely, leading to a problem of novel class detection. In cases where it is impossible to know the entirety of the domain we are dealing with, closed set methods cannot generalize well enough to model classes unobserved at training time and prove insufficient to deal with concept discovery and model sparsity, since observed patterns from a class may change or new concepts may emerge (Gama, Žliobaitė, Bifet, Pechenizkiy, & Bouchachia, 2014; Parker & Khan, 2015; Masud, Gao, Khan, Han, & Thuraisingham, 2010). Moreover, when dealing with streaming data, multiple problems require attention. In the first place, we cannot save the entire stream as practical memory issues would arise implying that an abstraction of the data is required. It should, as accurately as possible, capture the distribution of the observed domain. Secondly, a learning method should be able to update, using just newly observed samples and a representation of the previously observed data, also without the need of annotated data. In cases where data is generated at a large scale, annotation or prior information regarding the data may be unavailable and thus the model should be able to structure observations in an interpretable way, depending on the application.

Traditionally, solutions for novel class discovery are proposed based on clustering-based approaches with the general premiss that samples close to each other in a generic space share more similarity than ones further apart, although this principle fails in the high dimensional scenario where sparsity is much higher.

One can look beyond this group of methods to others that provide actual characterization of the statistical distribution of the data. With the work of Rudd, Jain, Scheirer, and Boult (2017), both statistical information from the data and methods to deal with an unknown num-

ber of classes are used. The authors' goal is to perform image recognition (multiclass) in a open set environment using open world decision boundaries, where these are used to separate know classes from the unknown space effectively attempting to label samples from an underserved class as such. This approach is based on extreme value theory (Coles, Bawa, Trenner, & Dorazio, 2001), that dictates the form of the functions for the radial probability of inclusion of a point with the respect of the class of another. The training set corresponds to a set of extreme vectors (EV) that are related to the radial inclusion function modelling the probability of sample inclusion. The notion of open risk space was introduced by Scheirer, de Rezende Rocha, Sapkota, and Boult (2012) and consists on the risk associated with labelling data far from known training examples. The concept of selecting the points and distributions that best summarize each class, i.e., are least redundant with respect to one another, the authors arrive to a probabilistic representation of the class's decision boundary characterized in terms of its extreme vectors (EV), which provides an abating bound on open space risk, where data points further away from the boundary are less likely to belong to a given class. Regarding their experimental setup, they run the benchmarks with original setting on multi-class open set recognition on the Letter dataset and open world recognition on ImageNet.

The authors' approach is promising although it only labels unknown data as such, not incorporating it in as a new class. Doing so, requires a retraining of the model.

Work in the topic of novel class detection has also been done in the field of signal processing. Gharghabi et al. (2019) propose a domain-agnostic online segmentation model for multidimensional time series. In this work, time series extracted from motion sensors, for example, are analysed with the goal of identifying meaningful regime changes along a time series, such as detection of the transitions between walking and running or of certain patterns in heart rate. Hence, the semantics captured is shaped in the form of discrete classes. To achieve this, the authors use similarity-join metric for time series. It receives a time series T as input and a subsequence of length L, representing the size of the pattern, and returns two vectors. The first corresponds to the Euclidean distance between the subsequence and its nearest neighbour elsewhere in T (defined as MPValue). The second indicates the location of each of the nearest neighbour of each element of the subsequence in the time series T (defined as MPIndex). These two vectors lead to an annotated time series where one can derive the likelihood of a regime change.

By leveraging the above aspects, the authors propose FLOSS. This method is composed of two key components. An arc represents the i^{th} entry in the MPIndex indicating the nearest neighbour for the element at i^{th} location. The second component, the Arc Curve (AC), is an annotated version of the original time series where its i^{th} index specifies how many nearest-neighbour cross-over location i, i.e., how likely it is to be a regime change. If, at a location i, the number of arcs that cross over is small, this indicates a change of regime. Furthermore, since they are attacking an online problem, they use a sliding window where the Arc curve is computed for that given segment. To solve the problem of updating the minimum distances of the subsequence of the sliding window, when a point leaves it, the arc computation is done only from elements further in the time series to ones that occur earlier than them.

Despite the difference in domain, the work described contains some similarities with ours. In both cases there is a search for semantically relevant segments although, in this case, the information extracted and the types of events that are detected are of much lower level than the ones we are modelling. It is relevant to consider if FLOSS can be used in our case, either directly from the audio signal, or with some of the features previously described. Capturing local similarity in music is not a new approach by any means although the modifications taken into account in this paper for the model to work in an online setting are prone to consideration.

Gjoreski and Roggen (2017) also focus on the discovery of activities such as running, walking or jumping, characterized by sensor signals. Their approach is based on agglomerative clustering and aims at exploiting the temporal information in the signal. The methodology consists on, at a given point in time, keeping a number of active clusters estimated by clustering the frames of a given time window, so that multiple deviations can be clustered into multiple temporally overlapping segments. The total number of clusters in the active pool does not represent the total number of clusters, which is open ended. Each of these clusters has a tolerance that gives the duration the cluster is allowed to exist without being updated (merged or deleted), with the goal of modelling short outliers and a minimum duration that discards the cluster if it was only present for a short period. Although it approaches a similar problem, the parametrization used includes defining the number of active clusters at any given time as well as the tolerance and minimum duration times. For our work, defining the number of active clusters in a time window is not desirable as its too restrictive, given that prior knowledge about the expected number of clusters is not available.

There as also been work done on bringing the modelling capabilities of neural networks, specifically CNNs, into the field of incremental learning and novel class detection. This class of models lack the robustness to deal with novel classes, due to the assumptions of closed world datasets with a fixed number of categories. The work of Z. Wang, Kong, Changra, Tao, and Khan (2019) focuses on addressing this challenge by leaning a feature representation such that distribution of instances from the same class are discriminative enough in order to perform label prediction, novel class detection, and subsequent model adaptation.

Their model is divided into three main components, a step that aims at transforming the samples into a subspace where samples from the same class are closer to exemplars of each class, a second step where novel-class instances are detected and a third step that updates the model either by adding new classes or by updating existing ones. More specifically, the first step is to train a network to transform the observed data points such that they are close to a set of prototypes for each known class. This object is defined as a tuple $p =< \mu, d, w, \xi >$ containing the mean of the network for a set of inputs $x_i \in \xi$, μ, d corresponds to the sum of squared Euclidean distances to μ, w is the size of ξ and ξ is a set of data points from the same class. Because this method is designed to solve an incremental learning problem, the prototypes are updated while the network is training, so that the final prototypes maximize the distance between ones of different classes and minimize the distance to transformed data points.

The second step focuses on novelty class detection. For this step, due to the assumption of non-stationary data, the authors assume a Gaussian distribution for each known class and use this to compute a statistic of the confidence of a data point belonging to a given class. If, for a given threshold, the new data point is rejected by all classes, this data point is inserted in a buffer. When the buffer is full, the true labels for the points are requested and the model is incrementally updated, by retraining with the new data points, including the estimation of the prototypes for each of the new classes observed. The authors argue that this process, despite being slow, is minimized by the small size of the buffer, as the incremental update of the network and prototype estimations are only done for a small number of samples at a time.

The main issue with this article is the use of true labels in the retraining step, in order to update the classifier, as in our case, that type of information is not available. Moreover, there is access to labelled data at training time, that will act as prior knowledge for the incremental

updates, again not available in our setup.

Serra, Müller, Grosche, and Arcos (2014) propose a method for music structural annotation using time series structured features and segment similarity. They aim at annotating the structure of a music piece in an unsupervised way without employing explicit knowledge of previously annotated pieces, by detecting temporal locations of segment boundaries and to assess segment similarity based on repetitions. They achieve this by building a model that firstly extracts tonal and harmonic features from the audio. They then transform these into a time series of structured features from which they compute a novelty function whose peaks correspond to boundaries. Finally, the resulting segments are compared in a pairwise fashion and clustered.

More specifically, the information of each sample in the time series is improved by incorporating information from the sample's recent past. This is achieved using delayed coordinates where a sample x_i is constructed following:

$$\hat{x} = [x_i^T x_{i-\tau}^T \cdots x_{(m-1)\tau}^T]^T \tag{2.6}$$

Eq. (6) corresponds to the concatenation of the feature at timestep i, by the previous features down to $(m-1)\tau$ where τ corresponds to a time delay and m is the total amount of information being considered for the task (the dimensionality of the $\hat{x}$). To assess the homogeneity (passages of music that are consistent with some musical property such as rhythm, timbre or harmony) and repetitions a recurrence plot is computed. It consists of a square matrix R whose elements indicate pairwise similarity between samples. A subsequent step involves the creation of structural features. The authors represent the homogeneity and recurrences of R in a time-lag matrix where correlation is measured between each sample and samples increasingly further away in the time series, both past and future in the case of this article. This is done for the purpose of incorporating homogeneity and repetition with correlation between samples. The time-lag matrix is considered as a sample from a bi-variate distribution and in turn this distribution represents the probability mass-function of time-lag recurrences. To approximate this distribution, the time-lab matrix is convolved with a bi-variate rectangular Gaussian kernel. This Gaussian kernel is computed as the product of two Gaussian windows corresponding to the lag and time dimensions. These windows will influence how many columns are needed

to represent the convolved time-lag distribution.

The estimated distribution can be seen as a time series along the time axis and structure features are then defined as columns vectors. These encapsulate both homogeneity and repetition, from the recurrence plot, as well as robustness against time and lag variations, due to the kernel convolution. Finally, boundary detection is done by computing differences between successive structure features. These values yield a one-dimension novelty curve where peaks correspond to values above a given threshold and, at the same time, correspond to a global maximum of a given window. Segment repetition is then evaluated using Q_{max} measure (Serra, Serra, & Andrzejak, 2009). It is a generic and configurable time series similarity measure that exploits the information contained in the traces of a recurrence plot.

The work described here is related to our own with the key distinction that an online setting is not considered nor structure has the same meaning as in our case. The encoding of similarity between data points should be embedded in the feature itself but can not benefit from statistics done using the entire data set, as it is the case of the article described.

Work that resembles our own from the musicological perceptive is the one from (Krause, Zalkow, Zalkow, Weiß, & Müller, 2020). In this paper, the authors conduct a case study on a dataset covering 16 recorded performances of Wagner's Ring of Nibelung, with annotations of ten central leitmotifs. They build a neural network classification model and evaluate its ability to generalize across different performances and leitmotif occurrences. These motifs constitute the classes of the classification task. Furthermore, all motif occurrences were annotated by a musicologist. In terms of the classification task, the authors define it problem of assigning a given audio excerpt to a class according to the occurring leitmotif, discarding segments where multiple occurrences of different leitmotifs happen in parallel.

A final key point in this work is the dual approach to the classification problem. The first, the performance perspective, concerns variabilities across different performances, resulting from different instrumental timbres, tempi, or other decisions made by the artists, that can lead to the album effect. The second perceptive looks at the compositional or occurrence concerns in regards to the diverse musical variabilities of leitmotif occurrences in the score. These two perspectives motivate the authors two perform two distinct slips on their corpus, the first, based on the performance, select the three recordings with manually annotated measure positions for the test set and three performances with automatically transferred measure positions for

the validation set. The remaining ten performances are used for training. In this split, all subsets comprise all occurrences of all motifs In contrast, for the occurrence split, they randomly choose 80 of the occurrences for training and 10 each for the validation and test set.

The authors also study how temporal context (audio before and after the leitmotif sequence) affects classification and compare to a scenario where samples correspond to audio segments with a motif attached. Overall results are lower with the second data split described than for the first. Finally the authors introduce another setup where, due to the case where the score contains regions with other or with no leitmotifs at all. In this setup, they introduce a noise class to cover such scenarios. They show that despite the lower results, this class in not detrimental to the task.

This work shows major similarities with our regarding some of its difficulties and goals, specifically the identification of leitmotifs and the capability of producing a model that is able to generalize across multiple interpretations of the same motif or score. Despite these similarities, our task faces broader issues, as we do not posses such a fine grained annotation, nor the the multiple interpretations of the same score. Particularly, the leitmotif can suffer modifications through the material it supports, but further interpretations of these changes may help a learning model generalize each leitmotif better.

Effectively all the methods presented so far provide some insight on some of the methodologies available to us, either tackling the problem of novel class detection or the problem structural discovery. Although they face the same family of problems, them methods presented always make considerations regarding model adaptation, an incremental learning procedure or the use of annotated data, effectively circumventing one or more of these issues. To this effect, the following chapter present methods that can be directly used in our context, proposing solutions for the aforementioned problems simultaneously, without relaxing the problem in a way that allows to circumvent the issues present in our work.

2.3 Density Estimation

Mixture models can a be a powerful tool to model uncertainty. They allows us to represent specific parts of a domain. A mixture model can be seen as the weighed sum of individual

probabilistic functions and formally defined as:

$$p(x|\theta) = \sum_{i=1}^{K} \pi_i F(x|\theta_i) \tag{2.7}$$

where K is the number of components in the mixture, π_i is the weight of the component, θ_i is the set of parameters of the probability distribution and $F(.|\theta)$ is the probability distribution parametrized on θ. When using this model in an unsupervised setting, with the premiss that each components approximates a class present in the domain, each sample has only a given probability of belonging to each class, i.e., to a component of the mixture model.

There are two key aspects when using this model. First, there is a choice of the number of components of the mixture model. This number influences how well modelled a partition of the data is. Second there is a choice of the distribution F. How well the model is able to explain the data depends on this choice given the underlying shape of the distribution of the data. If the last is unknown, distributions that can model the variance in the observed data are recommended, hence the Gaussian or the Student-t distributions are good candidates.

One method for approximating a mixture model to a density function is Kernel density estimation (KDE). Generically, it is a method to estimate the probability density function of a random variable. It is defined as:

$$\hat{f}_h = \frac{1}{n} \sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{2.8}$$

where K is defined as a non-negative function that integrates to one, usually modelled using a Normal density function. h corresponds to the smoothing parameter or bandwidth, where higher values may lead to over-smoothing.

Based on the described concepts, Kristan et al. (2011) introduce the multivariate online kernel density estimation method. Their goal is to approximate the distribution of the data, explained by a GMM, given an online setting where samples, after observed and processed, are discarded. The authors call this GMM the sample model. The proposed model is based on two key points: the first is that the model is non-parametric in the sense that the number of components is unknown a priori and can can grow given the observations. The second point is that each new observation corresponds to a Dirac-delta function and, during online operation,

each new sample is added to the sample model in the form a new component.

Because samples are assumed to be Dirac-deltas, if the no processing is done, the number of components grows linearly with the samples. This is unwanted, as little information expandability is had when this occurs. To make sure that the number of components is able to grow, but at the same rate as samples, a compression algorithm is used to approximate a number of samples by a single distribution. Each component of the used GMM fits a portion of the data and a KDE is then used to approximate the GMM to a probability distribution density function.

The compression algorithm consists of two steps. The first revitalises the mixture by splitting components that are no longer good approximations of the data. The second merges components that are sufficiently similar. Furthermore, as new data arrives, compressions preformed at a given time may later become invalid and so a detailed model is maintained for each component in the form of a two-component GMM. This model corresponds to the simplest possible model that allows recovery from the mentioned problem. An overview of the methodology can be seen in Figure 2.4.

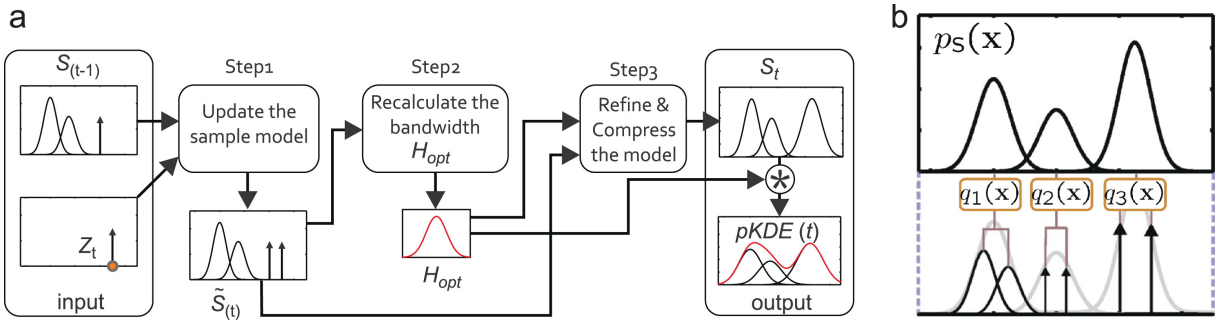


Figure 2.4: (a) Overview of online KDE iteration. The sample model S_{t-1} is updated with a new observation z_t. The optimal bandwidth is recalculated, and if the parameter allows, the compression routine is called. This leads to a new sample model S_t. (b) Illustration of the sample model S_t(sample distribution $p_S(x)$ along with the corresponding detailed models for each of the components. Adapted from Kristan et al. (2011).

Critically evaluating this model, we can see that the main driving force of the model is spatial location of the data. This dictates the number of components the model generates and if we consider changing the distribution of the mixture model, this will directly impact how well explained the data is.

Furthermore, the splinting and merging criteria are based on a distance measure and a predefined threshold and do not reflect underlying uncertainty of splitting said components.

Other rather relevant parameters we can modify, besides the choice of distribution, are the compression thresholds, as the similarities between data points may be general enough that a lower value can be used, resulting in a denser model, with a lower number of components. A forgetting factor, also proposed by the authors, can also be modified. It causes a higher weight to be given to younger samples than older ones when the compression routine is called. A different threshold for younger and older components can also be implemented so that older components are more resistant to change or outliers than younger ones.

There is a clear gain when using this model for our task. Since the number of leitmotifs is unknown a priori, the non-parametric nature of this model is capable of dealing with this aspect. Its explainability through the sample-model also proves adequate as the data is approximated through distributions. Another important point is that we want the number of components in the mixture model to approximate the number of leitmotifs in the musical stream, which we assume to correspond to different classes. The KDE approach gives us no guarantee of this approximation as the number of components may grow as much as needed in order to produce a more accurate explanation of the data.

2.4 Dirichlet Process

The KDE approach faced the unknown number of clusters through density estimation. In the case where we want to approximate the number of components in the mixture to the number of classes (leitmotifs), we propose the use of the Dirichelet process and Bayesian machinery. This model explicitly models the uncertainty of creating new clusters, in our case, new components. These have shown good results in other domains such as the cases of topic modelling and robotics (Nakamura, Ando, Nagai, & Kaneko, 2015; Nishihara, Nakamura, & Nagai, 2016).

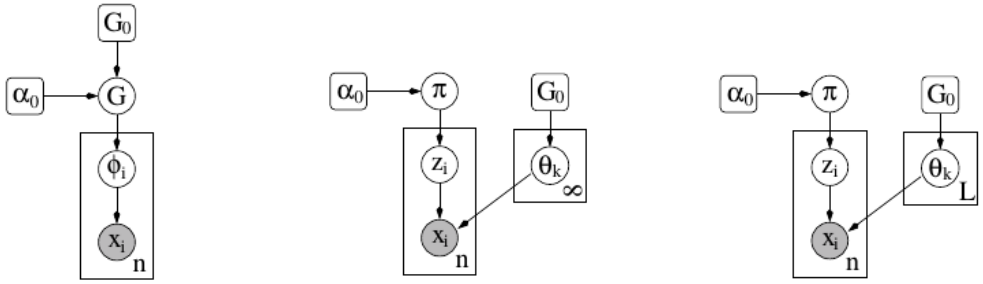
The Dirichelet process (DP) is a member of the family of non-parametric stochastic processes. Let (Θ, β) be a measurable space, with G_0 a probability measure on that space. A Dirichelet process DP(α_0, G_0) is a distribution of a random probability measure G over (Θ, β), where α_0 is a positive number, such that for any finite measurable partition of Θ, the random vector $(G_0(A_1), \ldots, G_0(A_r))$ is distributed as a finite-dimensional Dirichelet distribution with parameters $(\alpha_0 G_0(A_1), \ldots, \alpha_0 G_0(A_r))$ Eq. (9):

$$(G_0(A_1), \ldots, G_0(A_r)) \sim Dir(\alpha_0 G_0(A_1), \ldots, \alpha_0 G_0(A_r)). \tag{2.9}$$

If G is a random probability measure with distribution given by the DP, then it can be written as $G \sim \text{DP}(\alpha_0, G_0)$ where G_0 is a base measure that can be seen as a prior guess of the data (Antoniak, 1974). α_0 corresponds to the concentration parameter that gives the degree of belief on G_0 (Ferguson, 1973). The choice of base distribution will have a great impact on the model performance and is guided by mathematical and practical convenience leading to a choice that is conjugate with the underlying model, improving both computation time and model simplicity (Görür & Rasmussen, 2010). Each draw from $\text{DP}(\alpha_0, G_0)$ delivers an infinite object G that can be written as Eq. (10):

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k} \tag{2.10}$$

where δ_{θ_k} is a probability measure concentrated at location θ_k with a weight π_k. This characterization of G is one of the possible forms of describing an infinite mixture model as represented in Figure 5.6a.



(a) (b) (c)

Figure 2.5: (a) A representation of a Dirichlet process mixture model as a graphical model. (b) An equivalent representation of a Dirichlet process mixture model in terms of the stick-breaking construction. (c) A finite mixture model representation. Adapted from Teh et al. (2006).

So far, the DP was presented as a theoretical mathematical object. One important question that we must answer is how do we know that such an object exists and how we can construct and represent a DP. Teh et al. (2006); Hjort, Holmes, Müller, and Walker (2010); C. E. Rasmussen (2000) present three such constructions where they show the existence of the DP: one based on the stick breaking construction, one based on a Pólya urn model(equivalent to the Chinese restaurant process (CRP)) and one based on a limit of finite mixture models. The various representations of the DP are mathematically equivalent but their formulation differs because they examine the problem from different points of view. We focus on the CRP which provides a sim-

ple and computationally efficient way to construct inference algorithms for Dirichlet Process.

The CRP is a preferential attachment model that directly reflects the clustering of draws from the DP. It is defined as a distribution over partitions and is explained by the following metaphor. Consider a Chinese restaurant with an unbounded number of tables. When the first customer arrives, he can randomly select one empty table (cluster), sit and order one dish. Then, the second customer can either join with the first customer and share the dish, or he can start a new table and order a new dish. In this way, when the n_{th} customer arrives, he can select one table from k occupied tables with probability proportional to the number of guests, m_k already seated there, or start a new table with probability proportional to α_0. Formally, the conditional probability can be written as :

$$\text{CRP}(\theta_n|\theta_1, \theta_2, \ldots, \theta_{n-1}) = \begin{cases} \frac{m_k}{n-1+\alpha} & \text{if } \theta_k \text{ exists} \\ \frac{\alpha}{n-1+\alpha} & \text{if } \theta_k \text{ is new} \end{cases} \tag{2.11}$$

In this metaphor, the tables correspond to clusters and the dishes correspond to the parameters of the distribution of each cluster.

This can be directly applied to a mixture model as a non-parametric prior distribution on the components of the mixture model. Each table is a draw from G and we can define ϕ_i to be the prior of the parameters of the distribution of a component of the mixture and can be written as Eq (12):

$$\begin{aligned} \phi_i \mid G &\sim G \\ x_i \mid \phi_i &\sim F(\phi_i) \end{aligned} \tag{2.12}$$

where $F(\phi_i)$ denotes the distribution of the observation xi parametrized by ϕ_i. ϕ_i is conditionally independent given G, and takes on values θ_i, following Eq. (11). This specification gives us a method to increase the number of clusters while modelling the uncertainty of doing so.

Despite giving us a good method to grow the number of clusters in the data, the model comes with some drawbacks that can be circumvented. Firstly, sampling methods are required to estimate the predictive distribution and, as a closed form is not obtainable, its computation may result in a large overhead depending on the inference method and its precision. Furthermore, Markov Chain Monte Carlo (MCMC) methods generally require the full set of data

points to function as it is the case of the standard Gibbs sampling method. The MCMC family of methods corresponds to one of the most widely used methods for inference in the CRP as the problem of finding the optimal cluster assignments translates into adding and removing mass to each of the clusters (Neal, 2000). This becomes prohibitive in our setting, leading us to use inference methods that can function in an online setting without maintaining all the samples in memory.

Another issue we face is these models' dependency on initialization, in the form of priors, and since we have no prior information about the data in question, an update of the prior belief must be done throughout time, solely based on the observed data.

A final remark we must point out is the inability of the model to model temporal information given the baseline definition we gave. Because exchangeability of the data is assumed when defining the CRP, the model, in its general form, does not account for the order of the data, although it can produce clustering assignments in streaming data, in cases where sequence of observations is not relevant. To face this issue, for scenarios where modelling such information is relevant, (Guo & Gong, 2017) propose a method to add temporal information to the cluster estimation of the CRP by replacing m_k in equation 2.11, that represents a count, with the aggregate influence of the sample at timestep k. This is obtained from the samples in the sliding window that have the same cluster assignment, i.e., the guest in the restaurant that sit at the same table. This influence is computed via kernel density estimation of the samples contained in the sliding window. In their work, the authors use a dynamic time window since they aim at topic detection in social media where the amount of data per time step can increase with the number of users. In our case, the sampling frequency from the audio file is constant and so we can use a pre-defined window size.

Another extension to the DP is the Hierarchical Dirichelt Process (HDP) (Teh, Jordan, Beal, & Blei, 2005). With the DP, draws from G_0 are independent and identically distributed (i.i.d.) and so components do not share samples. In a context where this type of modelling is desired, such as the case of topic modelling, where we have multiple documents and want a distribution of topics across documents, we might want to model the shared weight of topics across different documents. This is achieved by sampling G_0 itself from a DP. This hierarchy may have as many levels as the ones required for a specific task.

In cases where the distribution of classes follows a particular pattern, extensions to the DP

have been proposed to deal with data in these conditions. The Pitman-Yor process (PY) is a two-parameter generalization of the DP that leads to heavier-tailed, power law distributions for the frequencies of observed objects or topics (Pitman, Yor, et al., 1997; Teh, 2006; Sudderth & Jordan, 2009). It is defined as $PY(d, \alpha_0, G_0)$ where d is a discount parameter between 0 and 1 and the rest of the parameters are as as described earlier. Going back to the metaphor of the Chinese restaurant, the discount parameter benefits the creation of more tables and, as d grows to one, that chance increases. It is expected to observe a higher number of tables, but with less costumers in them. The conditional probability can then be modified from Eq. (2.11) as Eq. (13):

$$\mathrm{CRP}(\theta_n|\theta_1, \theta_2, \ldots, \theta_{n-1}) = \begin{cases} \frac{m_k - d}{n-1+\alpha} & \text{if } \theta_k \text{ exists} \\ \frac{\alpha + d|k|}{n-1+\alpha} & \text{if } \theta_k \text{ is new} \end{cases} \tag{2.13}$$

where $|k|$ corresponds to the number of tables already occupied by one or more costumers.

As mentioned in Section 2, the distribution of leitmotifs is not uniform. Regarding film narrative, more emphasis is made on the main characters or themes of the movie than on secondary plots, that occur around the main ones. This is directly reflected on the film score, where leitmotifs for the kernels of the narrative will have more occurrences, approximating a power law behaviour. Because PY-process has been shown to produce clusters that follow this type of behaviour, this extension of the DP may prove more accurate for our problem.

There is a need to compute the posteriors of the proposed Bayesian models. Because this work is set in an online setting, inference methods that require the full set of points in order to converge to an optimum, be it local or global, or that require iterating threw all the observed points, are not available and so other alternatives must be studied.

The first mention is due to the streaming nature of the data we are analysing and the second point is justified by the finite amount of memory we posses leading to the inability to store all previous observations.

(L. Wang & Dunson, 2011; Crook, Gatto, & Kirk, 2018) propose an alternative to MCMC, which allows approximate Bayes inference under one DP mixture by performing sequential updates. Given the CRP setting, a new sample is allocated based on cluster that maximizes the conditional probability of that sample belonging to a given cluster and it is assigned to a new cluster if a statistic about the distribution of the data votes higher than any of the clusters. There

is a clear trade-off between accurately estimating the predictive distribution and the speed in we which they do so. This method much is faster than MCMC approaches although it requires some prior information to build the initial statistics and is dependent on the permutation of the data. Because of its speed the authors propose to run the model with different reshuffles of the data and choose an ordering that maximizes the pseudo-marginal likelihood. For our context this can not be implemented as we want to preserve information regarding the sequence, leading us to run the inference process with just the sequence of observations from the musical stream.

Another common approach in the literature for inference in the Bayesian setting is Variational inference. In variational inference, we specify a family Q of densities over the latent variables. Each $q(z) \in Q$ is a candidate approximation to the exact conditional. The goal is to find the best candidate, the one closest in Kullback-Leibler divergence (KL) to the exact conditional. It turns the problem of approximate inference into a problem of optimization, in order to find the optimized member of Q, q^*.

However KL is written as Eq. (14):

$$\mathrm{KL}(q(z)||p(z|x)) = \mathrm{E}[\log q(z)] - \mathrm{E}[\log p(z,x)] + \log p(x) \tag{2.14}$$

where $x = x_{1:n}$ is a set of observed variables and $z = z_{1:m}$ is a set of latent variables, with joint density $p(z,x)$, and where the term $\log p(x)$ is intractable. Because computing the KL is not possible, an alternative object that is equivalent to the KL up to an added constant is optimized instead:

$$\mathrm{ELBO}(q) = \mathrm{E}[\log p(z,x)] - \mathrm{E}[\log q(z)] \tag{2.15}$$

It corresponds to the negative of the KL from Eq (14), plus $\log p(x)$, which is constant with respect to $q(z)$. We can then state that minimizing the KL is equivalent to maximizing the ELBO function. Examining the ELBO gives intuitions about the optimal variational inference. The ELBO can be rewritten as as a sum of the expected log-likelihood of the data and the KL divergence between the prior $p(z)$ and $q(z)$:

$$\mathrm{ELBO}(q) = \mathrm{E}[\log p(x|z)] - \mathrm{KL}(q(z)||p(z)) \tag{2.16}$$

The first term is an expected likelihood. It encourages densities that place their mass on configurations of the latent variables that explain the observed data. The second term is the negative divergence between the variational density and the prior, encouraging densities close to the prior. Thus, the variational objective mirrors the usual balance between likelihood and prior.

Having described the ELBO, the variational objective function that will find q^*, it is required to describe the the variational family Q. A commonly used family is mean field variational family. It assumes the latent variables are mutually independent and each governed by a distinct factor in the variational density following:

$$q(z) = \prod_{j=1}^{m} q_j(z_j) \tag{2.17}$$

Each latent variable z_j is governed by its own variational factor, the density $q_j(z_j)$. In optimization, these variational factors are chosen to maximize the ELBO, Eq. (15).

Having both the specifications of the ELBO and of one the variational families that can be used. The final step corresponds to an algorithm to solve the optimization problem. For the specific context of DP mixture models, work in the literature as been done for adapting the generic model just described into the particular setting of the DP. We will follow the work of Tank, Foti, and Fox (2015) and Huynh and Phung (2017) in order to implement this type of approximate inference in our problem. Both these methods propose variational inference methods and optimization algorithms for cases of streaming data while using the DP.

2.5 Language-based Methods

Another approach that can benefit us is the use of n-gram models. Sequences of musical structures, either low level structures, like frames or notes, or higher level structures like chords or segments, contain additional information because they happen close to each other and more importantly, in sequence. With the premise that sequences of observations carry additional information, we are able to derive a symbolic representation of the audio and use these in the online context. The level at which we build the sequence will heavily impact what our model is learning. For example, sequences of MFCC frames will model changes in timbre along a short period of time.

More specifically, methods derived from topic modelling and language models, in the context of HDP have been used for speech segmentation. Raczyński and Vincent (2014) propose a genre dependent topic model, for modelling chords that aims at predicting a genre of a music using a distribution of chords.

Work has as also been done in the field of word segmentation from phoneme sequences by Takeda, Komatani, and Rudnicky (2018). This work aims at building systems that can acquire knowledge during their spoken interactions with human beings. Unknown or new words can frequently appear even if we carefully prepare a vocabulary set in advance. To combat this problem, the authors propose a model based on subword N-grams and subword estimation using a vocabulary set, and posterior fusion of the estimation results of a Pitman-Yor semi-Markov model (PYSMM) and their model. The PYSMM integrates both word-level and character- (phoneme) level N-gram language models and then estimates the segmentation labels of phonemes corresponding to word boundaries by updating both language models in an unsupervised manner. A subword refers to a unit smaller than a word. If subword patterns of vocabulary words are obtained from a given vocabulary set that does not include duplicated words, then the subword N-gram model can capture better "word-level" segmentation patterns as words than a phoneme Ngram model. Their proposed vocabulary model estimates the subword pattern of a word in an unsupervised manner. Although the subword model is superior for detecting out of vocabulary words, it might degrade the sentence-level segmentation accuracy. The estimation results of a PYSMM and their model are merged to take advantage of both.

2.6 Summary

In this chapter, we reviewed and compared state-of-the-art models and systems that relate to our task. We began by referring the most common approaches to feature extraction for musical audio. Segmentation approaches were then mentioned as way of splinting the audio based on the underlying features and work in the literature related to extracting and or grouping information given a signal was covered.

We presented the background for the methods considered experimentally in our work. We began by introducing the KDE method, an unsupervised online approach based on density

estimation, that allows us to cluster data with an unbound number of groups. Much like it, the family of the DP methods provide the same set of benefits, albeit, through a completely distinct method of estimation. The final section of the chapter referred to language based methods as possible approach to model and cluster feature data at a higher level.

3 Dataset Preparation

The aim of this chapter is to cover pre-processing steps required for the construction and preparation of the source materials associated with audiovisual content analysed in this work. Data was extracted from multiple sources, specifically, the audio from the movies themselves, their scripts, the subtitles, the chapter information and finally their respective soundtracks. We therefore do not have a single dataset, but a collection of distinct elements that make up the material related to the movies.

We begin by presenting the dataset, followed by work towards obtaining the musical audio played during the movies. The subsequent section then approaches how we obtain narrative characterization of the events in the movie at any given time and that can characterize the audio that is being played, from that point of view.

3.1 Dataset

We used the Complete Recordings of the movie adaptation of Tolkien's The Lord of the Rings, by Peter Jackson, containing the complete score for the extended versions of the films. The Lord of the Rings score, composed by Howard Shore, accompanies almost entirely the films, where each track was produced for a given segment of the movie with a thematic background emphasizing how the movie tells the story, therefore enriching it. The score was selected because of the extensive work that has been done in the past analysing its compositional, structural, cultural, and literature background. It was produced solely for the movies, taking inspiration from the source material, the books. It offers around 13 hours of composed music that provide substantial data to work with.

Because of the extensive literature available, concept discovery that is done on this music can be interpreted with contextual story and cultural background, directly bridging the image with the musicological aspects. This aspect will allows to compare the quality of the struc-

ture created by our learning methods with the one agreed upon by the literature, giving us a validation tool.

Howard Shore uses similar thematic material through all his work on the trilogy, leading to an opportunity to study not only how the leitmotifs are used and related to each other but also to study how do these relate to the visual, emotional and cultural aspects shown in the movies, an analysis than has been done (not in a computation setting) by Young (2007); Adams (2010). The composer took inspiration from the descriptions that are present in the books, mainly the in depth descriptions of the inhabitants, the instruments used in each region (where each region is associated to a fictional culture, that has different leitmotifs associated with) as well as poetry that is sung by the characters, that show the importance of music within Tolkien's novels. There is also effort put into the novels, in order to deeply characterize the world which also leads to greater cultural background, later used to compose the soundtrack. For example, the exert "Doom, doom came the drum-beat and the wall shook ... Another harsh horn-call and shrill cries rang out" is depicted in the movie and is accompanied by a musical rich in drum sounds showing a clear inspiration from Tolkien's descriptions.

The music of each of the races in the fantasy world contrasts in terms of instruments, pitch, and melody. These abrupt contrasts allows us to better isolate and capture each leitmotif of a given culture because of the very distinct features.

Another relevant feature in Shore's work is the presence of voice in some of the music produced. It is restricted to some domains as it is the case of the Elven music or motifs related with Sauron where there is a predominance of choirs. This feature further helps us to structure the different motifs as the presence of voice is descriptive for specific ones.

Songs like "The Shire", are played throughout the three instalments although with different instrumentation, style, harmony, and melody while maintaining the same base structure. One occasion where we can observe this difference is when Shore uses a flute to carry the violin melody line of the theme when ever Frodo, a character of the series, is reminisces about the Shire, correlating this theme, in particular, with the character by changing the instrumentation used for that particular motif. Another example is the "horn-Shire" theme, a more heroic version of the song, played with a French horn, that relates to the transformation of the hobbit characters. The motifs contained in each of the versions of the same songs differ due to their specific use through out the three movies, while at the same time keeping and underlying

similarity that can be described in terms of a hierarchy, where the base contains the constant motifs and further branches explain specific modifications to the main theme. Cases like the one described are one of the main objects of study for this work. There is a clear progression in terms of the leitmotif composition through time, where leitmotifs introduced early on branch into different ones to better follow the narrative that is being told.

3.2 Audio Preparation

For the dataset used, we have access of two versions of the audio data. The first corresponds to the movie's soundtrack. The second corresponds to the music that is played in the movies, that for editing purposes, motivated by driving the story forward or other creative reasons, does not correspond directly to the music in the official soundtrack. The music present in the movie suffers distortions in terms of energy in scenes where the dialogue is to be more emphasized, for example, as well as being accompanied by other sound effects, such as character dialogue, battle scenes or world events. All these aspects contribute to a decrease in musical audio quality when working with the music present in the movies, compared to the use of the audio from the soundtrack. These aspects motivated us to find mechanism to increase the audio quality of music in the movies.

An initial approach to this problem led us to experiment with source separation tools with the goal of isolating the musical audio from the the other audio components. However, as we were unable to successfully isolate the musical audio, we resorted to a solution based on audio alignment. This choice was made so that we can retrieve a high quality audio version of the music that is being played throughout the movies.

The tool implemented, based on Dynamic time warping (DTW), (Sakoe & Chiba, 1978), takes 20 second audio fragments from the movie and aligns it with the highest score audio fragment of the same length from the soundtrack. The algorithm's score is computed taking as input the chromagram of from the audio segment from the movie and the set of 20 second fragments from each track in the soundtrack. A semi exhaustive search is conducted to find the highest alignment score (a skip of 500ms was implemented to decrease the search space). This is repeated for each movie so that less comparisons have to be made.

For this dataset, prior information that the music present in the movies was played in an

order that was respected in the soundtrack. Taking this into account, an heuristic was included when choosing the aligning segment from the obtained alinement's rank. A percentage of the total time of a track must be aligned before a segment from another track can be chosen.

With the alignment process established, each 20 second segment of the movie has a corresponding segment of audio from the soundtrack associated with it. Nonetheless, it is important to point out that this alignment is not perfect. Because of the noise that comes associated with the music in the movie, this can, in many cases, distort the shape of the feature we extract to perform the alignment, therefore negativity affecting it. The numeric values, metrics and features described where chosen based on empirical evidence, such that the audio alignment would be as accurate as possible.

3.3 Ground Truth and Metadata

We set as goal to capture relationships between narrative events that are similar in nature, given their musical audio counterpart. It is important to define what these event are and how we group them together, as these become the ground truth information from where we derive our conclusions. For this purpose, this section approaches how we can derive a ground truth from the metadata that accompanies the movies. Moreover, the approach that will be described is not bound to our dataset. As long as the required material is available, this extraction process can be applied to other audiovisual content.

We begin by extracting speaker and location information, for every instance through the movie, from the scripts and subtitles that are part of our dataset. This was achieved with the use of a tool for subtitle and script alignment, originally developed by Rosado (2016). For the alignment of the script with the subtitles, they use the Needleman–Wunsch DP algorithm, (Needleman & Wunsch, 1970). The algorithm finds an optimal path between two sequences, and then, detects an optimal alignment between them. To use the algorithm with the script and subtitles, first, the script's dialogue and the subtitle's dialogue are tokenized into words, and then, a similarity matrix is created to compare whether or not each word is the same. After the alignment, if the number of words matched between any two sentences is more than 50%, then those sentences are considered to be equivalent. An example of the aliment being computed can be seen in Figure 3.1.

Since subtitles are time-indexed, any alignments with the subtitles stream is implicitly shared with other time-indexed data, such as audio and video streams. This gives us a tool to automatically retrieve pertinent information from the movie scrip. Originally, the tool would only output the speaker for each line of dialogue but throughout this work, it was extended so that the location information present in the script could also be retrieved. For clarification, speaker information corresponds to the name of the characters that are speaking and the locations correspond the the fictional places present in the scenes of the movie.

It is important to mention that, as this process is automated, it suffers from algorithmic errors that affect the quality of the produced match. This in turn, affects the quality of the ground truth produced. Through testing of the tool, it was concluded that it's output had a high recall per alignment, meaning that there is a correct alignment between the subtitles and the script, although the tool misses some of the matches resulting in an empty alignment, which implicates that there are some characters or locations that may not appear as frequently due to this error in alignment.

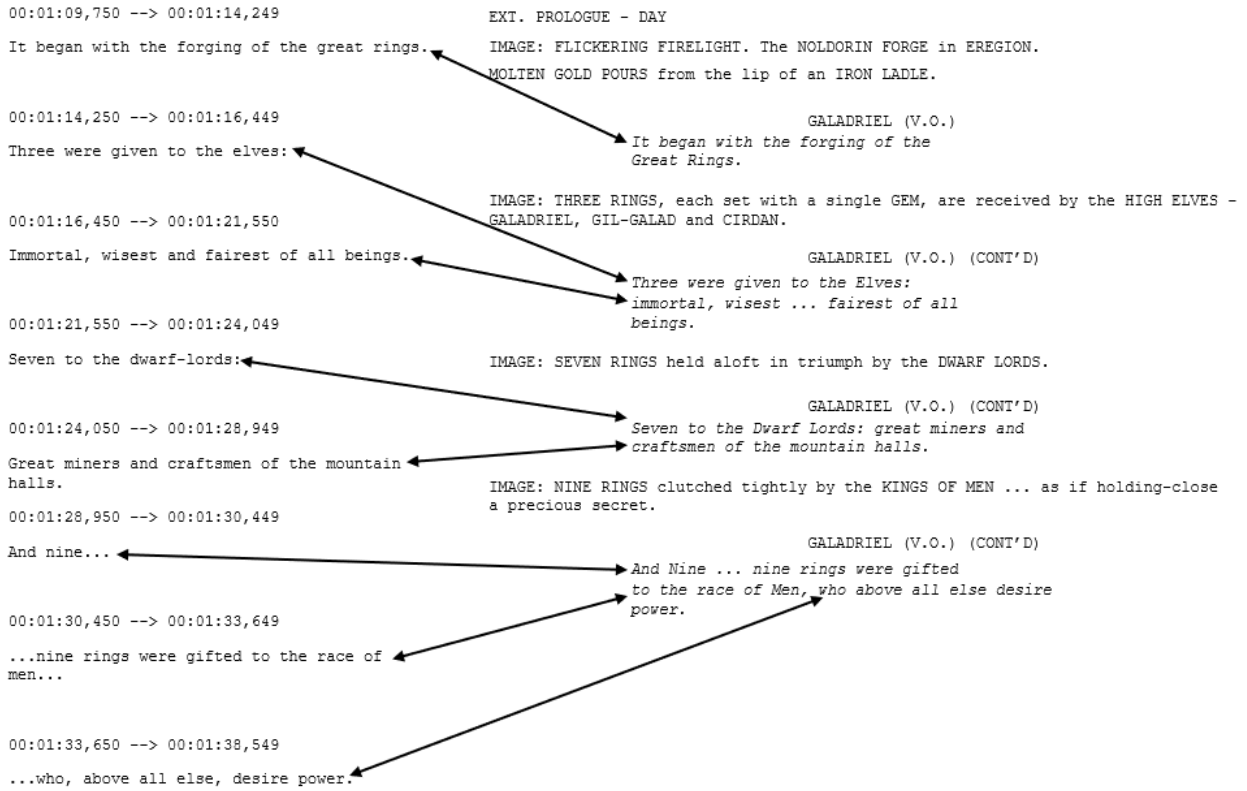


Figure 3.1: Example of alignment between subtitles and script.

Another key point in the construction of our ground truth is the use of **scenes**. Originally

introduced by Lopes (2017), these are defined as the sequence of utterance or lines of a single speaker and their use is aimed towards capturing the dialogue (consisting of one or more subtitles) of a single character or set of characters that are very close in time, based on the timestamps of the subtitles. We based the scene construction criteria on the previous work, as the authors expensively tested which values would fit scene creation best. The values used were 500ms as the maximum distance between two subtitles than belong in the same scene.

Rather than associating character and location at the subtitle level, to the corresponding music segment, we use the scene as a way of aggregating this information. This higher abstraction level, allowing for the association of a longer piece of audio.

Having the metadata information associated with each scene, another key decision was scene aggregation based on either the repetition of the exact same character(s) in distinct adjacent scenes or the repetition of location in the same conditions, therefore establishing two distinct ways of grouping information, each giving more weight to their specific aggregation key (character or location).

The decision to aggregate scenes was based on the knowledge that there are pieces of audio playing in the movie were the are no speakers present, implying that there is no annotation available for these segments. In order to leverage the annotation present per scene to other segments not covered by any subtitle and subsequently by any scene, we aggregate them as mentioned above, so that segments of audio between scenes that share the same metadata information may also share that annotation, thus increasing the overall amount of music that has ground truth information associated with it.

Given the established aggregation possibilities, we opted towards aggregating scenes by characters, as described above. We found this solution to be a good balance in terms of granularity of ground truth information being grouped. Aggregation by location resolved in very long sets of scenes with different characters, as the narrative in the movie can take place in a single location for a long period of time.

Finally, we point out a processing step in regards to location information present in the ground truth. The locations present in the script and subsequently in the annotation correspond to geographical locations from the fictional world. Due to the observation of fine grained locations present through the scripts, that we considered to be of too much detail, we opted to encase some of these locations into broader corresponding ones, always referring to the map

to make such decisions. Moreover, it is important to note that this change was had-doc for this particular dataset and possible to the existence of a detailed map of the world, that allowed us to make informed decisions. Such change is possible for other datasets as long as geographical information is present. In the case of the movies analysed, a map of the fictional world was first introduced in Tolkien's books and then later adapted for the movies, which we show in Figure 3.2. A list of the final locations used can be seen in Table 3.1.

	Locations
1	TOWER OF CIRITH UNGOL
2	WEATHERHILLS
3	MAP
4	EPHEL DÚATH
5	EDORAS
6	FANGORN
7	HELM'S DEEP
8	ISENGARD
9	BREE
10	DIMHOLT
11	MORIA
12	PROLOGUE
13	THE MISTY MOUNTAINS
14	PELENNOR FIELDS
15	ROHAN
16	THE GREY HAVENS
17	SHELOB'S TUNNEL
18	PATHS OF THE DEAD CAVERN
19	WHITE MOUNTAINS
20	ITHILIEN
21	MINAS MORGUL
22	MORDOR
23	EMYN MUIL
24	HOBBITON
25	MINAS TIRITH
26	XXX-EAST
27	RIVENDELL
28	OSGILIATH
29	XXX–SOUTH
30	DUNHARROW
31	SHIRE
32	LOTHLÓRIEN

Table 3.1: List of locations selected as abstractions from finer grained ones.

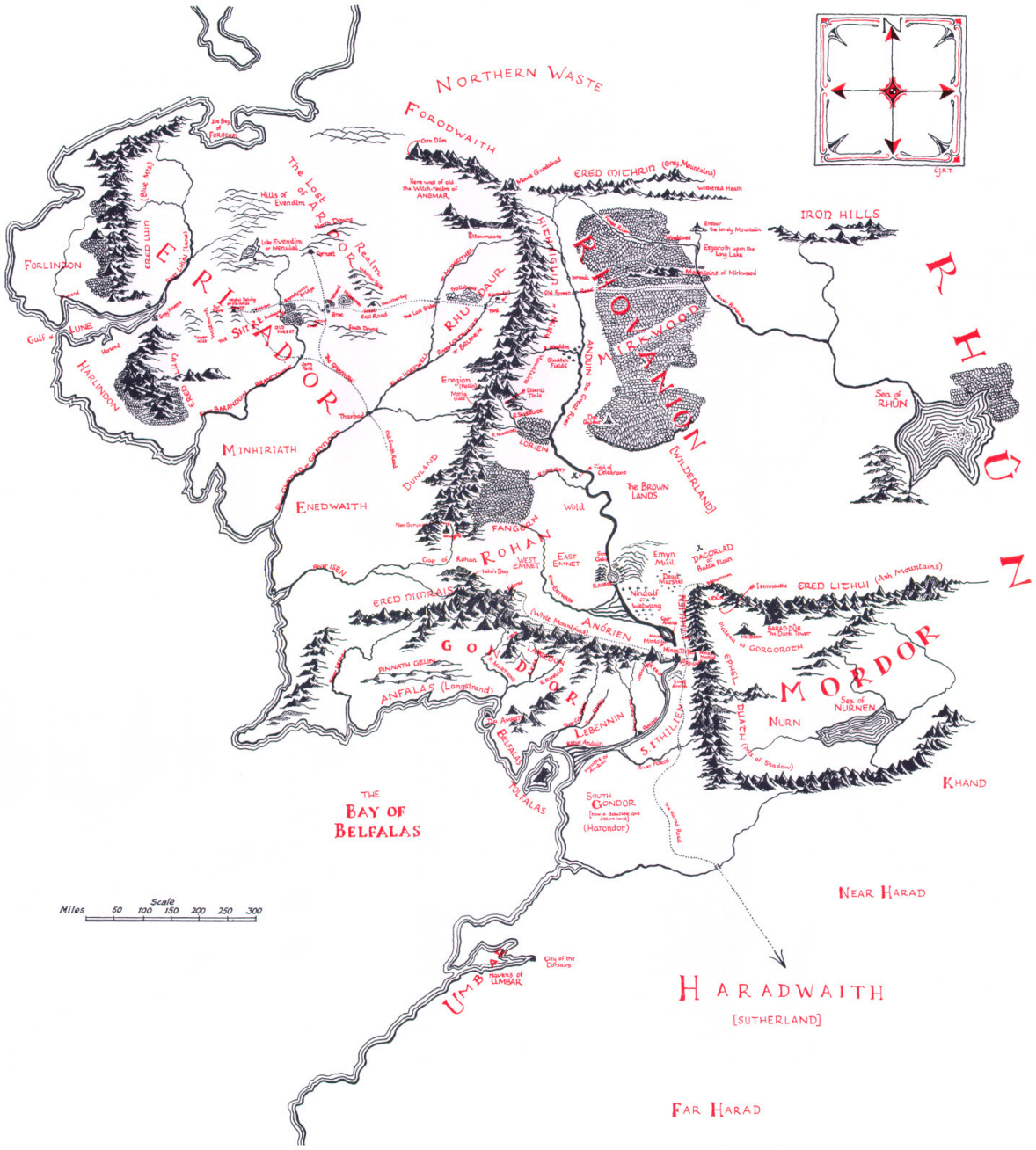


Figure 3.2: Map of the fictional world of the Lord of the Rings. From Tolkien (1991).

3.4 Clustering Labels

This processing pipeline, described previously, leaves us with a set of metadata information that is directly associated with each audio instance, i.e., for each set of aggregated scenes, we have the corresponding music that is played in that time interval. Specifically, location and character information. The two elements combined help characterize the narrative of the movie at any given moment, thus being possible to view these as elements that characterize a situation in the movie. There can then be an association between the audio and any of these items, either one-to-one or one-to-many.

The leitmotifs and musical audio in general suffer modifications through the narrative. This implies that the audio that is associated with a character or a location is not uniform throughout the movies. For example, the music from Minas Tirith used in the first movie, where Gandalf is present is very different in theme from the one present in the third movie, with the same character. I can also be seen that the same pair charter-location hold different narrative meaning in these two occasions and because of this difference, the underlying audio also changes tone. These two aspects motivates to go beyond the association of a piece of audio to pair character-location, as we find this association insufficient.

To cover this issue, we follow the approach of Chollet (2016), where the author relies on matrix factorization to reduce the dimensionality of the target labels. This method makes use of co-occurrence of the target labels, projecting the high-dimensionality target vectors. Formally defining this technique, let M be the binary matrix of aggregated scenes I and labels L where $m_{ij} = 1$ if i_i contains label l_j and $m_{ij} = 0$ otherwise. We then use matrix M to compute the Pointwise Mutual Information Gain (PPMI) for the set of labels L, that we will denote as matrix X. Let L_i be the set of scenes associated with label l_i, the PPMI is defined as:

$$X(l_i, l_j) = max\left(0, \log\frac{P(L_i, L_j)}{P(L_i), P(L_j)}\right) \tag{3.1}$$

where $P(L_i, L_j) = |L_i \cap L_j|/|I|$ and $P(L_i) = |L_i|/|I|$. Intuitively, the PPMI gives us the association measure between a pair of discrete outcomes x and y. In our case it measures the association between aggregated scenes and a context by calculating the log of the ratio between their joint probability and their marginal probabilities.

X is then factorized using Singular Value Decomposition (SVD) in the form $X \approx U\Sigma V$. Let Σ_d be the diagonal matrix containing the the top d singular values, and let U_d be the matrix obtained from selecting the corresponding d columns from U, we build the matrix $C_d = U_d \cdot \sqrt{\Sigma_d}$ that corresponds to the label factors in d dimensions. The item factors are obtained in similar fashion defined by the matrix $F_d = M^T \cdot C_d$. These two matrices encode the aggregated scenes and labels in the same projected space, respectively. Thus, a distance measure can be used to aggregate scene embeddings and labels. Similar labels are grouped in space, and at the same time, scenes with similar sets of labels are close together. In Figure 3.3, we can how both matrices project in a common space.

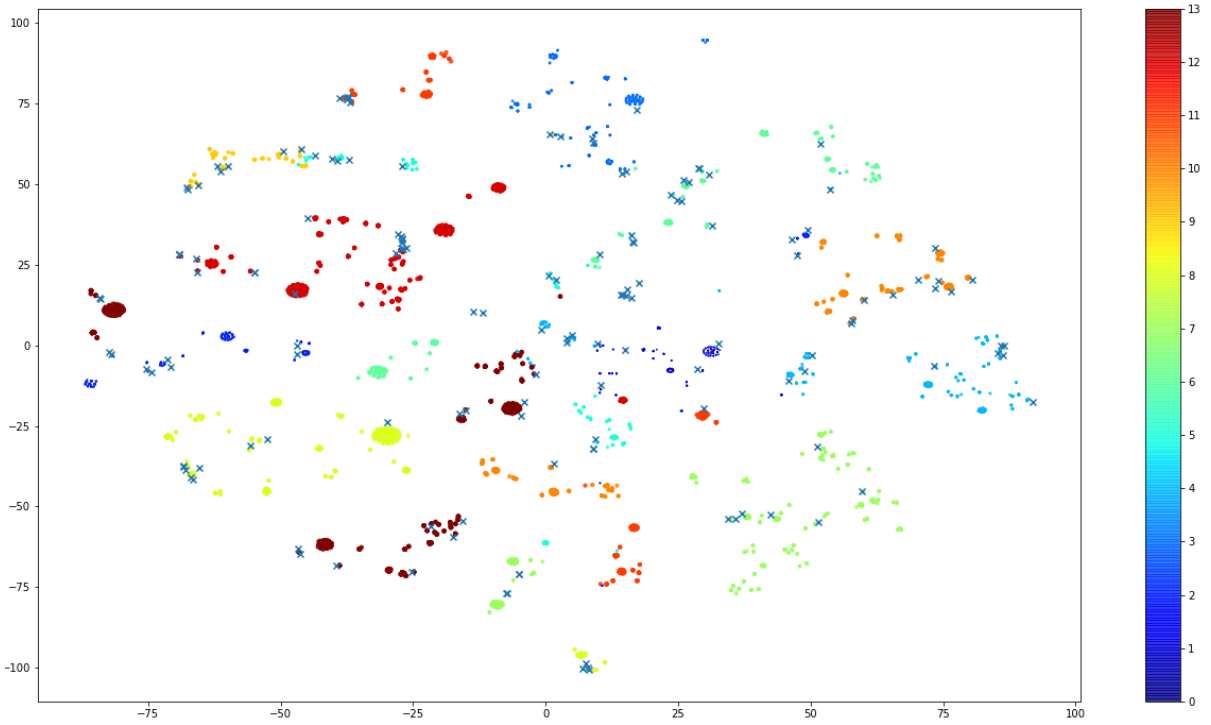


Figure 3.3: Crosses correspond to elements of the matrix C_d. Dots correspond to items of matrix F_d and are coloured given the cluster centroids they are closer to.

Clustering can be done in two distinct manners. The first is done by clustering aggregated scene embeddings, corresponding to the rows of matrix F_d. Aggregations of these items can be viewed as which aggregated scenes share similar labels. The second and more interesting type of aggregation made possible in the cluster of label embeddings, belonging to the matrix C_d. These clusters represent the metadata information that co-occurs, hence, that can be related explicitly. In our work, these labels can correspond to locations, characters and temporal information in the form of movie chapters or a fixed temporal window. If one is to include

these types of information as labels, we can encode which different aspects of the narrative and merge them for a more meaningful representation of the narrative of the movies.

The introduction of the temporal window allows us to differentiate the same charters at the same locations in different periods of the movie. This differentiation is wanted as due to narrative aspects, the audio being played, in the same region can evolve pushing us to make this type of distinction. Thought development, we experimented with the introduction of time via chapter information and by fixed size temporal segments. We concluded experimentally that 10 minute window segments created clusters where the items present were more compatible. Moreover, this type of information is common to segment the narrative of movie. In disk format of distribution of movie, it is common to have the notion of chapter information built in. In the Lord of the Rings movies, these average around five minutes, length that we found to be too short. By having temporal segments greater that chapter length we effectively grouping this information.

Finally, the quality of the clusters produced is focal for the rest of this work. The number and shape of the label clusters is very affected by the method used, thus it requires attention. Clustering can be performed directly on the items of matrix F_d or on the labels of matrix C_d. By doing this, we explicitly use the label embeddings to group aggregated scenes, as these are only implicitly shaped by label co-occurrence.

Two approaches were considered for clustering the labels on the matrix C_d. The first was to use KNN (k-nearest neighbours algorithm) with a number of clusters chosen empirically. This corresponds to our hard clustering approach, in the sense that the only metric in question is the distance between the label points. Because of these points, it presents two clear limitations: the choice of k, the number of clusters, and the absence of modelling variance or uncertainty of grouping labels together. To mitigate these limitations, a second clustering approach based on the KDE model was considered. By training one KDE with label data, the number of clusters grows has needed in order to build better explanation of the underlying data. Like wise, has the mixture components that make up are Gaussian, the uncertainty of grouping labels is being taken into account when building the clusters.

As the clusters were built using matrix C_d, it was necessary to cluster the items of matrix F_d given the computed sets of labels. In the case of KNN model, the centroids were used to label each item from F_d. In the KDE case, we assigned the cluster whose component yield the

highest likelihood.

Regarding the clusters obtained with KDE model, we found that the number of clusters was correlated with the dimensionality d of the SVD decomposition. The number of clusters grew the higher the dimensionality. Moreover, label clusters that were more representative of overall number of aggregated scenes preset in the movie presented a higher weight in the corresponding mixture component. This relationship can be seen in Table 3.2.

Cluster ID	**Weight**	**Time (minutes)**
0	0.152318	64.25
1	0.0596026	23.916
2	0.0529801	19.5
3	0.0794702	11.583
4	0.0529801	8.25
5	0.0264901	7.75
6	0.0596026	8.416
7	0.0331126	23.916
8	0.0397351	28.666
9	0.0596026	20.833
10	0.0794702	38.333
11	0.0397351	52.0
12	0.0198675	11.0
13	0.0463576	20.833
14	0.0331126	8.083
15	0.013245	4.25
16	0.0728477	33.833
17	0.0397351	13.083
18	0.0264901	11.333
19	0.00662252	0.916
20	0.00662252	8.5

Table 3.2: Mixture weights per component and corresponding coverage of each cluster.

As a final clustering approach, for the case on KNN clustering, we set the dimensionality of matrix factorization to 20, and number of clusters to 14. In the case of KDE approach, the dimensionality of the matrix factorization was set to 5, yielding 21 clusters. Their projections and coverage can be seen in Figures 3.4, 3.5, 3.6 and 3.7, respectively.

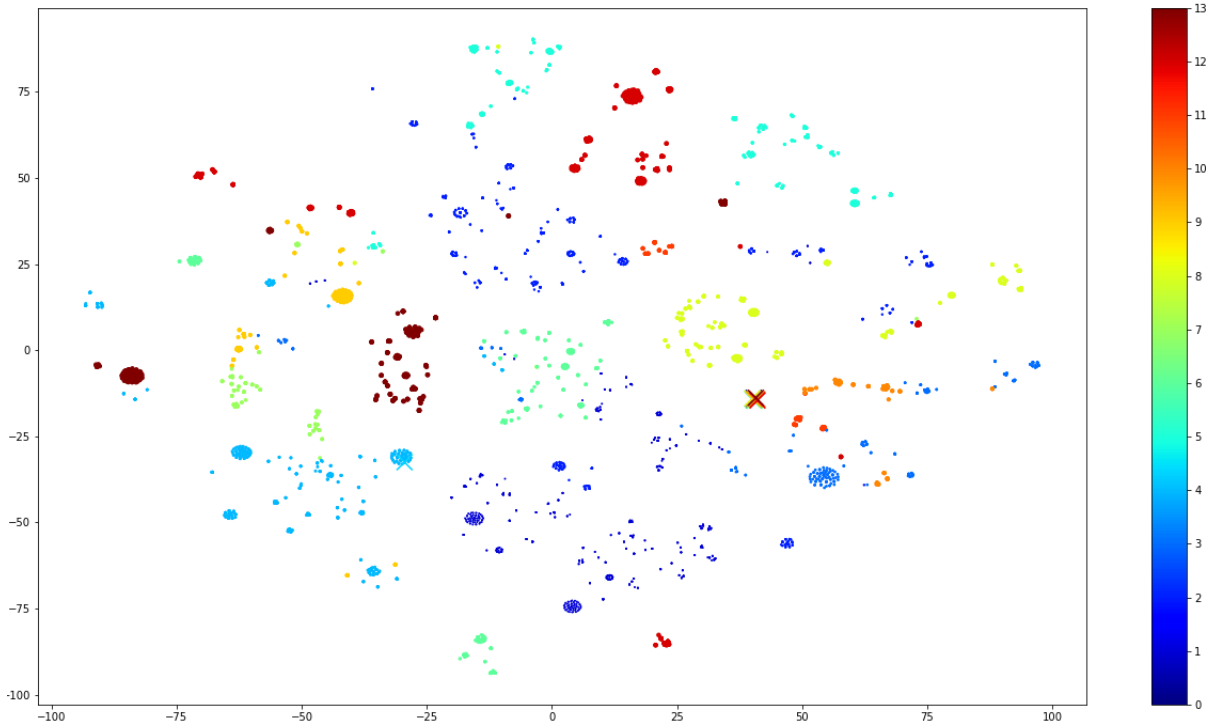


Figure 3.4: Projection of F_d items coloured by KNN clustering on C_d matrix.Crosses correspond to the centroids of the clusters obtained.

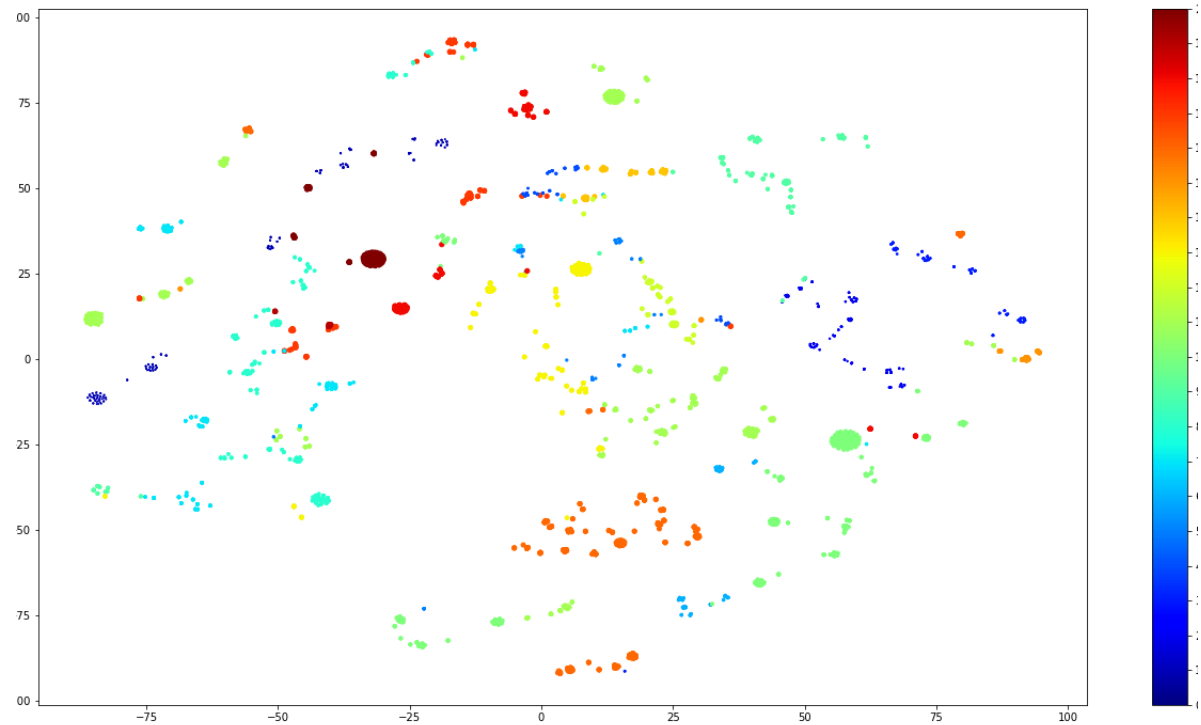


Figure 3.5: Projection of F_d items coloured by KDE clustering on matrix C_d.

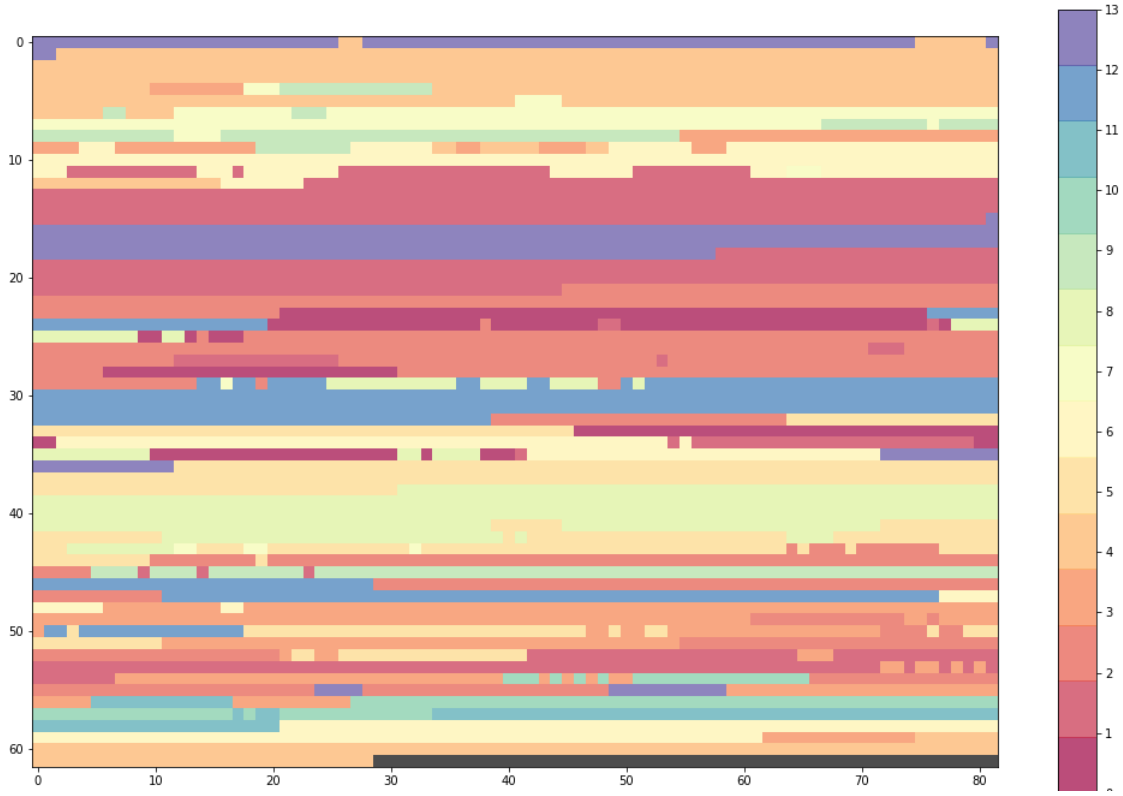


Figure 3.6: Coverage of each cluster through the movies timeline. Clusters were generated using KNN method.

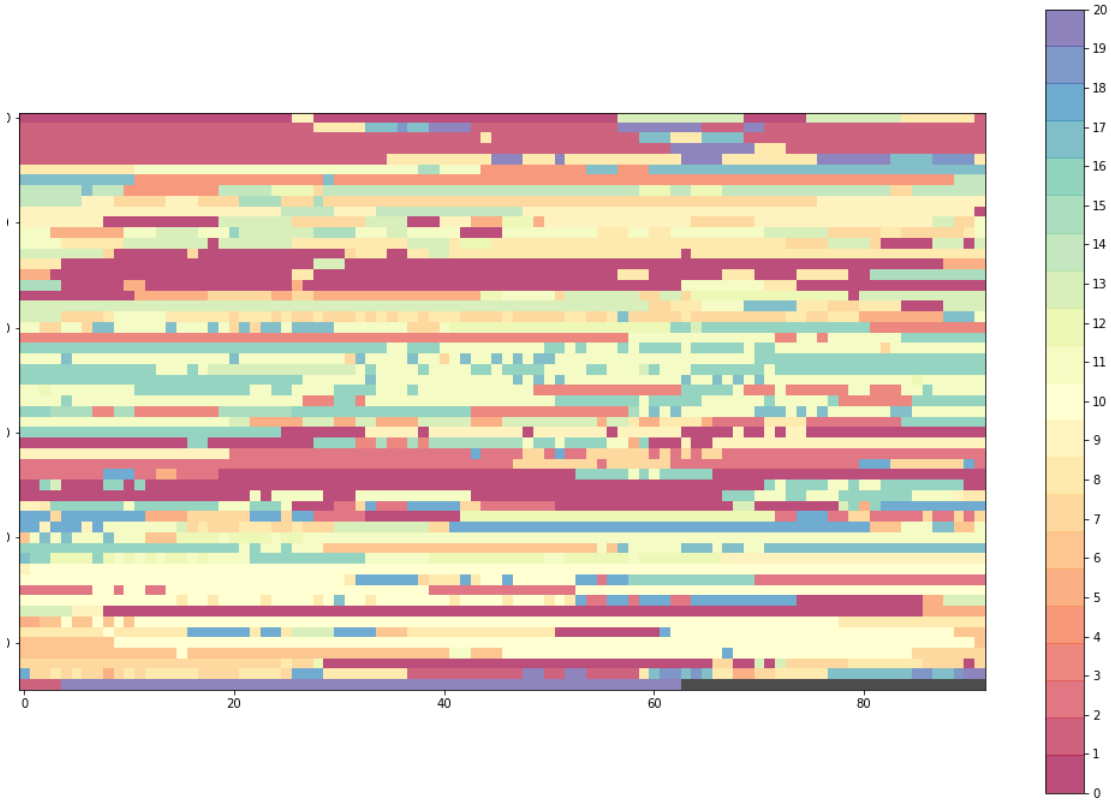


Figure 3.7: Coverage of each cluster through the movies timeline. Clusters were generated using KDE method.

3.5 Summary

We introduced the corpus used, consisting on the extended version of the Lord of the Rings movies, the corresponding soundtracks, scripts and subtitles. Moreover, we refer how to increase the quality of the musical audio from the movies by proposing an alignment method with the respective soundtrack. We propose a method to associate metadata information to musical segments from the movie by leveraging the alignment between the movie script and the it's subtitles, obtaining character and location information.

The concept of scene was introduced as a way of reducing the granularity of character and location information. A subsequent aggregation step was applied as way of of both correcting possible alignment errors between the scrip and the subtitles and as a way to expand the ground truth window to more music segments in the movie that may occur in between scenes. Temporal information is also added to the ground truth tag of each musical segment, so that we can differentiate characters and locations in different moments of the narrative.

Finally we propose how we can aggregate similar situations, by using a method of matrix factorization on the label space, in order to determine, based on the co- occurrence of characters and locations across different time periods, what sets of labels share similar narrative context. Two methods are used to solve this issue, one based on hard clustering using KNN algorithm and a approach using KDE method where the number of cluster is also outputted by the model.

4 Experimental Setup

In this chapter, we present our approach towards leaning relationships between situations with the same narrative context, in an unsupervised fashion, given the soundtrack that accompanies the movie narrative. We assume that different situations, defined in Chapter 3, as characters in a given location and point in time, because of the co occurrences of these elements, share similarities from the narrative point of view, thus sharing similar music. It is with the use of these similarities that we can map, depending on the perceptual features used, how two event scattered across the movies relate based on the audio that accompanies them.

In figure 4.1 a diagram of the pipeline of our work is showed. The last chapter covered the initial steps displayed, specifically the steps of music alignment, metadata extraction, and label clustering. This chapter covers the subsequent steps.

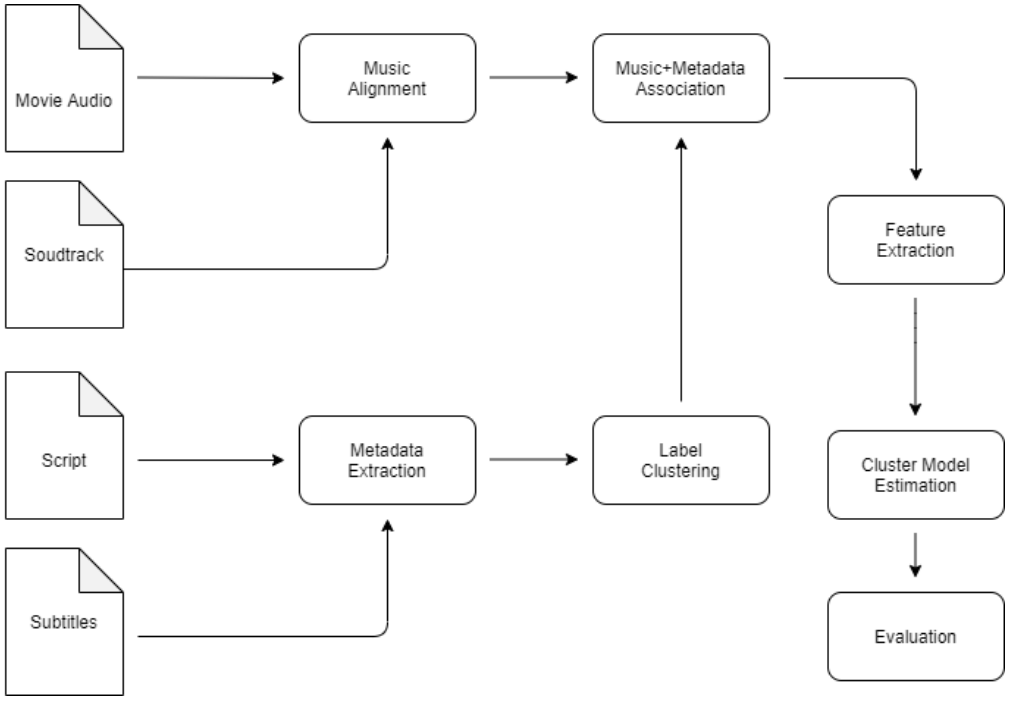


Figure 4.1: Diagram of the pipeline of our work.

The first part of this chapter covers the feature extraction step, necessary for all subsequent

work processing the audio features. The rest of the chapter covers our approach to the construction of the explanation of these audio relationships. To infer these, for each label cluster, we fit a model to the corresponding audio features in order to obtain a statistical representation of each cluster's audio. This is done in two different manners: 1 a baseline approach using multivariate Gaussian distributions; 2 A construction on the baseline using KDE model. Finally we describe how evaluation process was done.

4.1 Audio Setup

A pivotal choice in our work is the choice of which label clusters to use. The construction and analysis of audio relationships are built on top of these and so, the method used to generate the clusters must be weighted. Chapter 3 presented two alternatives for cluster computation. One based on hard clustering where the choice of the number of clusters was made empirically (so that the sets of labels inside the clusters and meaningful based on the knowledge of the movies present) and one obtained via an unsupervised method that accounts for variance of the labels present in each cluster. Based on these these two types of clustering approaches, the experiments presented here take into account both clustering scenarios.

Another important choice is how we chose to model the musical audio. In the last chapter, it was shown that the end result of of dataset preparation in terms of scene aggregation and it's respective audio were acoustic segments of distinct lengths with a set of narrative labels as a direct correspondence. These segments, in many cases, were too large to process, being in the order of minutes, aspect that motivated their split into smaller ones. By working with smaller sized segments, we aimed at capturing pieces of melodic content with fewer notes, which reduces both variability and complexity of the musicals segment and increases precision when extracting feature information. We set the size of each audio segment to 5 seconds.

The audio from each aggregated scene was then split into 5 second segments, with padding of zeros being added to segments that were smaller that the decided size. With audio segment size stipulated, all feature extraction and processing was done at this level.

4.2 Feature Extraction

The features selected were motivated by both the literature and the ability to distinguish, orchestral music in terms of timbre and tonal aspects. Specifically, the features extracted were MFCC, Chroma, chords, and a combination of MFCC+Chroma. Mfcc and chroma features were extracted with a fast Fourier transform (fft) window size of 2048, a hop length of 1024 between frames and a sampling rate of 22050 htz. Regarding MFCCs, the number of coefficients was set to 19: setting the number of coefficients to 20 and removing c0 coefficient (which indicates the average power of the input signal). In terms of the chroma feature, 8 octaves were considered and the number of "chromas" per octave was set to 12. Normalization was applied to the combination of Chroma and MFCC with respective variance, setting the mean to zero and variance to one, since the numerical range of each feature is different.

For chord extraction, we followed the approach of Müller et al. (2012). The first step, consisting of chormagram extraction, used the setup just described. The following steps generated, for each 5 second audio segment, the set of chords from the from the corresponding twelve major and twelve minor triads. As these correspond to discrete features, and to leverage additional information from the set of chords in each audio segment, we computed the TF-IDF (term frequency–inverse document frequency (Salton & McGill, 1986)). This gives us how important a chord is to an audio segment given the collection of segments available.

The aforementioned features, however, do not take into account melody information as they do not model sequence in any form. This issue was approached in two different manners. The first was to take the mean and variance of the frames from each five second segment. The second was to encode the audio sequence into an embedding with the use of an autoencoder tool. Regarding the first choice, the mean poses as a relevant mechanism to model the overall aspects of the audio segment. The variance was added in order to account for variation of information in the audio segment. Both were computed by fitting a multivariate Gaussian to each the set of frames for the segments being modelled. In terms of dimensionality, the diagonal of the covariance matrix of the fitted Gaussian is used and it's concatenated to the mean value of the feature for the given segment.

Regarding the encoding of audio segments, the approach previously described in Chapter 2 was followed. For this setup, for each five second audio segment, the mel-spectrogram was

extracted (with the same parametrization of the above features and 320 bins as suggested by the authors) and fed to the autoencoding network, with the base parametrization proposed. After training was complete, for each audio segment, the embedding was extracted from the hidden layer connecting the encode and decode layers.

4.3 Computing Relationships

Three different scenarios were considered in this work, using the features pointed above and the two types of clustering labels presented. As a baseline approach, for each label-cluster, we fit a multivariate Gaussian to the audio features associated with the cluster, thus obtaining the mean and variance values for the corresponding features, for each cluster. Relationships between clusters were computed as distance between the distributions that model the audio features of each cluster. We used the Bhattacharyya distance (Bhattacharyya, 1946) to compute this association. This procedure was implemented for each set of features presented above and with both types of label cluster generation.

A step up from these baselines was the replacement the Gaussian distribution with KDE model. By doing this, we are effectively changing how we build the audio explanation for each label cluster. To measure the distance between KDE model of each label cluster, we used the Hellinger distance (Hellinger, 1909).

4.4 Assessment

The evaluation procedure looked at the distance between the underlying models of each label cluster in order to assess if the distances reflect acknowledgeable relationships. For each setup, we computed a distance matrix between all label clusters and built a chord diagram. In the chord diagram, each entry corresponds to a label cluster and holds three outgoing edges corresponding to the top three most relevant relationships. The width of each edge is proportional to its relevance. Each edge was computed as the inverse of the distance between two label clusters, so that closer edges, i.e., ones that share stronger relationships, appear with wider edges in the chord diagram.

The work presented in this document is evaluated in a qualitative fashion, as it is challeng-

ing to quantify if the associations captured by the top are meaningful without a ground truth of said associations. It is because of this challenge that evaluation requires knowledge of the movies and score of the Lord of the Rings, in order to assess if the relationships captured are relevant. Furthermore, the assessment is also dependent on the features being used to characterize the audio of each cluster. Depending on these, one can look at the audio segments closest to the mean in the Gaussian case, or of higher probability function values, in the case of the KDE model, to complement the relationship assessment and to answer what each model is capturing. Therefore we also listen to these audio segments as part of our evaluation, to determine, depending on the features used, what is being grouped together.

An example of chord diagram analysis can be seen in Figure 4.2. The diagram displays a set of highlighted connections. These correspond to the most relevant relationships between cluster 3 and all other clusters. By consulting the appendix, we can decode the cluster by the most representative situations present in each cluster.

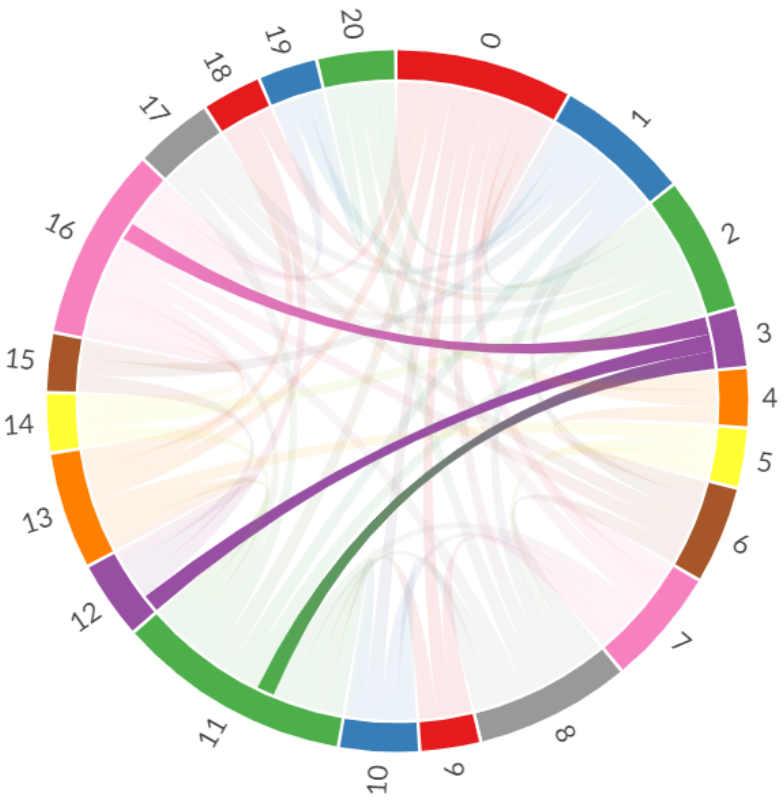


Figure 4.2: Top relationships computed between cluster 3 and all others. Thickness of edge represents more weight.

4.5 Summary

We presented how we built our experimental setup given all prior information regarding dataset preparation. We showed the two manners in which we cluster situations in the label space, based on the conclusions of the previous chapter. We justify the length of the audio segments used for all experiments. We then stated the details in parametrization in regards to feature extraction, and proposed two alternatives to capture sequence information, one implicitly based on a statistical distribution and one based on an autoencoding tool. Finally, we showed how the distances between clusters were computed, for both the baseline and the KDE approaches. The evaluation process was then described, focusing on the qualitative analysis of the relationships computed.

5 Experimental Results and Discussion

We present all the relevant experiments and corresponding results, highlighting the impact of using different features for our task as well as how changing the method used to model the underlying audio features of each label cluster influences the identification of relationships between these clusters. We begin my mentioning some of implementations necessary to build and evaluate the experimental setup.

5.1 Implementation

We focus this section on the mention of the different implementations necessary thorough this work. The tool for music alignment was built from scratch. It was necessary to implement all the logic regarding the processing of both the movie and soundtrack datasets as well as the heuristics for the alignment algorithm itself.

In regards to the tool for script and subtitle alignment, adaptations to the original tool was necessary, so that more metadata information could be extracted from the script, and to facilitate post processing of the information extracted, in terms of regular expressions. The synchronization of the higher quality audio segments, computed by the alignment tool, with metadata information, was a necessary step that had to be built as well. A set of regular expression rules was written, to transform the output of the scene-script aligner, into Pandas Dataframes, that facilitated integration with the corresponding audio segments.

The creation of clusters of labels was done as a direct implementation of the work of Chollet (2016), with both clustering methods being adapted to deal with the respective input.

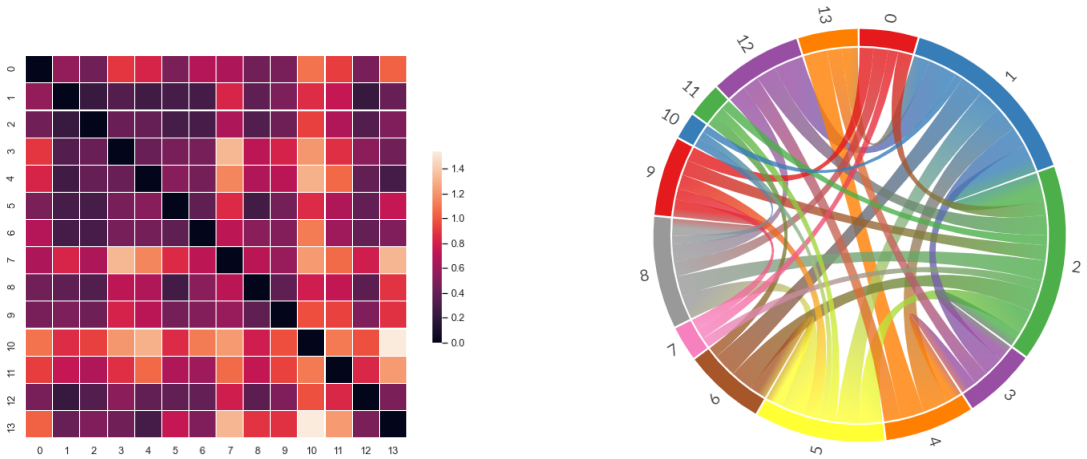
Concerning the methods used for feature extraction, the library Librosa, (McFee et al., 2015), was used. An exception to this was the adaptation of the work of Amiriparian et al. (2017), in the case of the production of sequence embeddings, so it could function with our dataset format, and the implementation of the chord extraction method, from Müller et al.

(2012).

Finally, the experiments described in the experimental setup were all built in the context of this work, as well as their evaluation. The KDE implementation was adapted so that we could, using the Hellinger distance, measure the distances between the different models. A implementation of the Bhattacharyya distance was also required to evaluate distances between multivariate Gaussian distributions.

5.2 Results

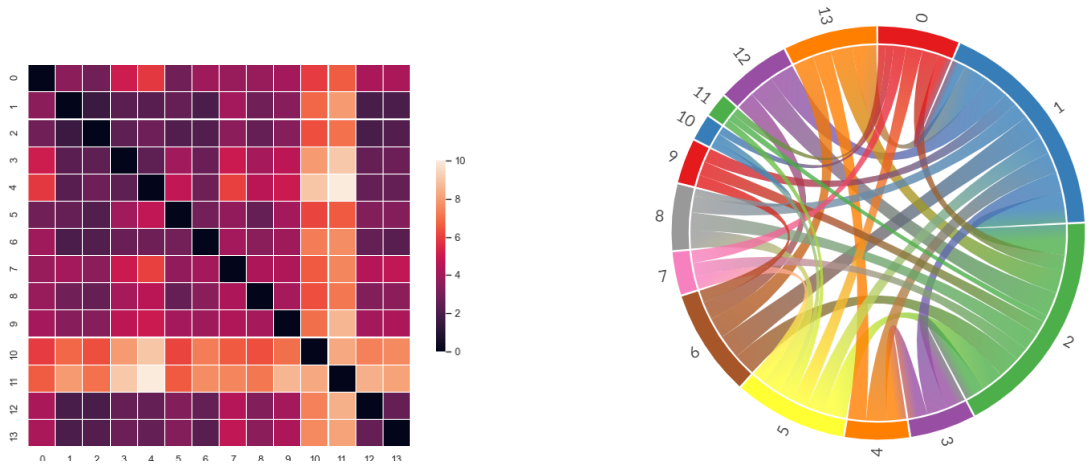
This section showcases the computed distance matrices and respective chord diagrams for the most relevant results, following the experimental setup previously described. We start by presenting some of the baseline scenario. First of all, Figure 5.1 shows a baseline experiment done using KNN clustering on the label side. The audio features of each label cluster, in this case Chorma, were modelled by fitting a multivariate Gaussian.



(a) (b)

Figure 5.1: (a) Distance matrix computed using chroma features and with KNN clustering on the label space. (b) Chord diagram of top relationships between clusters.

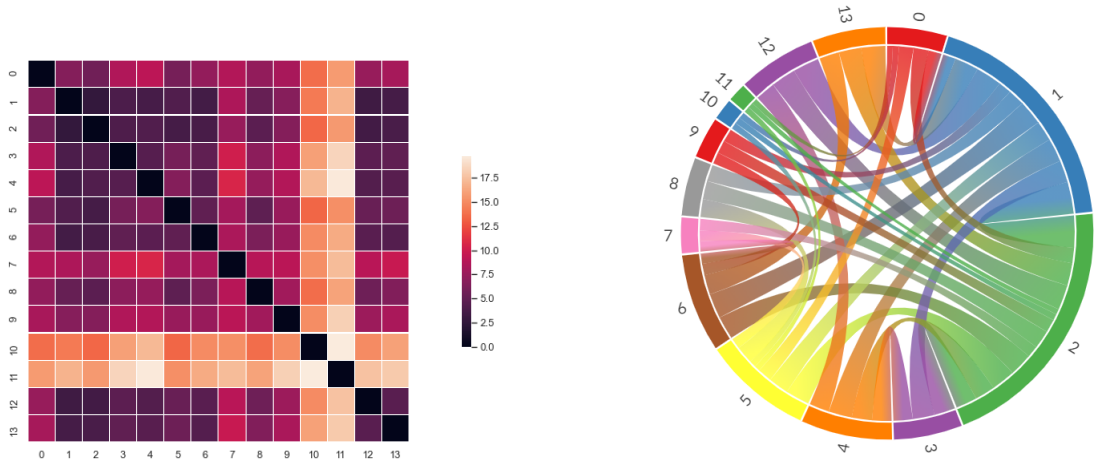
Figure 5.2 shows a second baseline experiment done using KNN clustering on the label side. The audio features of each label cluster, in this case MFCC, were modelled by fitting a multivariate Gaussian.



(a) (b)

Figure 5.2: (a) Distance matrix computed using MFCC features and with KNN clustering on the label space. (b) Chord diagram of top relationships between clusters.

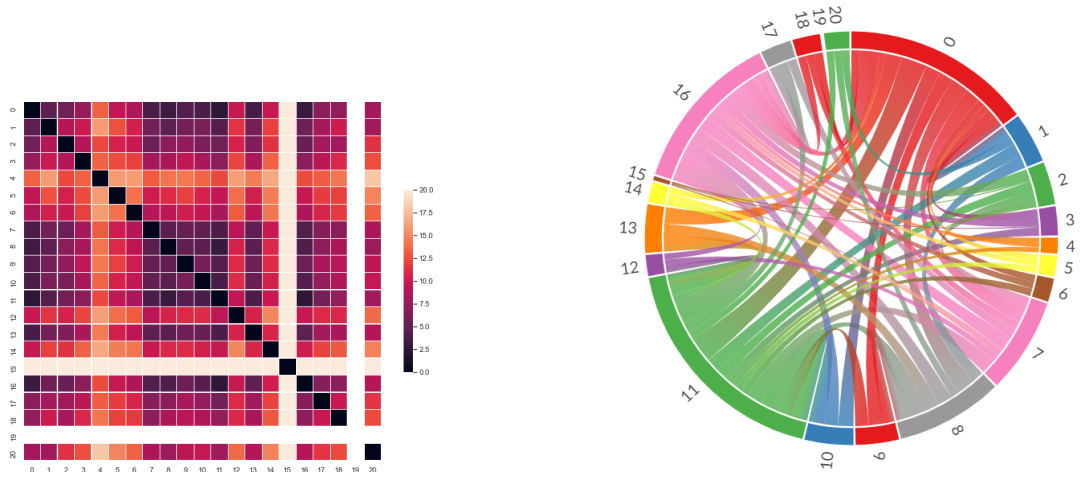
Figure 5.3 shows a third baseline experiment done using KNN clustering on the label side. The audio features of each label, in this case Chorma+MFCC, were modelled by fitting a multivariate Gaussian.



(a) (b)

Figure 5.3: (a) Distance matrix computed using Chroma+MFCC features and with KNN clustering on the label space. (b) Chord diagram of top relationships between clusters.

Figure 5.4 shows a increment to the other baselines experiments. The clustering on the label side was done using KDE clustering. The audio features of each label, in this case Chorma+MFCC, were modelled by fitting a multivariate Gaussian to the data of each label cluster.

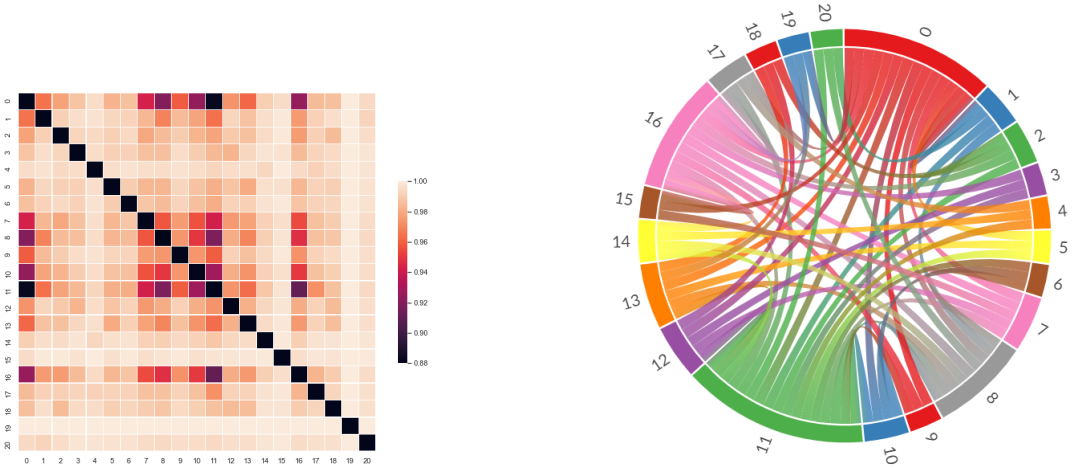


(a) (b)

Figure 5.4: (a) Distance matrix computed using Chroma+MFCC features and with KDE clustering on the label space. (b) Chord diagram of top relationships between clusters.

For the setups where KDE model is used for clustering in the label space and as a model of the audio of each cluster, the distance matrices do not contain the full scale of the Hellinger distance $(0-1)$. The choice to shorten the presented range was done so that relevant differences between clusters would be more evident.

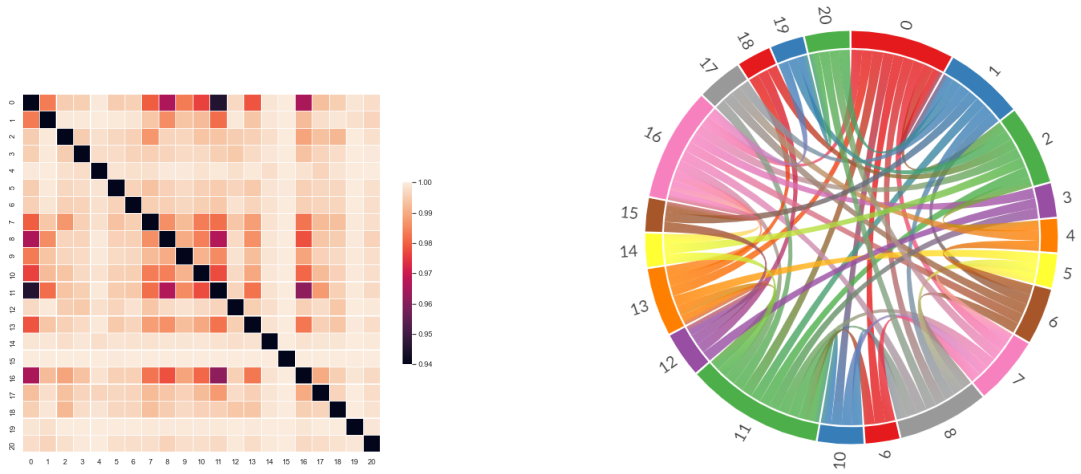
Figure 5.5 shows the first experiment done changing both the label cluster approach and the way the audio features of each cluster are modelled. KDE clustering was used on the label side. The audio features of each label, in this case Chorma+MFCC, were modelled by fitting a KDE model.



(a) (b)

Figure 5.5: (a) Distance matrix computed using Chroma+MFCC features and with KDE clustering on the label space and KDE used model the underlying audio of each cluster. (b) Chord diagram of top relationships between clusters.

Finally, Figure 5.6 shows an experiment done using KDE clustering on the label side. The audio features of each label, in this case MFCC, were modelled by fitting a KDE model.



(a) (b)

Figure 5.6: (a) Distance matrix computed using MFCC features and with KDE clustering on the label space, and KDE used model the underlying audio of each cluster . (b) Chord diagram of top relationships between clusters.

5.3 Discussion

It is always possible to compute distances between clusters and the challenge of this evaluation is to comprehend if these relationships in fact suggest that the music that is present in each of label clusters is able to relate similar situations across the movies. The presented results goal is to capture the differences that the clustering technique on the label side, together with how the audio from each cluster is modelled, help improve the ability to map these relationships.

We begin by looking at our most simple setup, where hard clustering is used to produce the label clusters and a multivariate Gaussian models the audio features in each one. The distance matrix shows us that the majority of the connections between label clusters do not differ greatly in size. Similarly, when using the KDE clusters and a Gaussian to model each cluster, the same behaviour is noticed. This lead us to conclude that increasing the complexity of the method used to create the ground truth is insufficient to obtain sufficient characterization of the label clusters.

The agglomeration of relations on some of the clusters, as it is the case of clusters 0, 11 and 16 in 5.4 can be explained by reading into the cluster composition. Although the situations that were aggregated share similarity from a narrative point of view, if the music range inside the cluster is too large, it will bring distribution that model the cluster closer to all others. As a concrete example, Figure 5.7 gives a higher insight into the content of cluster 0. The power law

behaviour observed was found amongst all clusters and is another indicator of the statement above. Cluster 0 presents elements from both the fellowship (Aragorn, Gimli, Legolas) and secondary characters that interact with them. These three character are predominant throughout the three movie instalments and the audio that is shared is varied in theme. The same point occurs in the case of Galadriel. The top occurrence, in the prologue location, contain music from different themes, including "The Ring", "The Ringwraiths", and "The Fellowship of the Ring", showing the vast variability inside a single label.

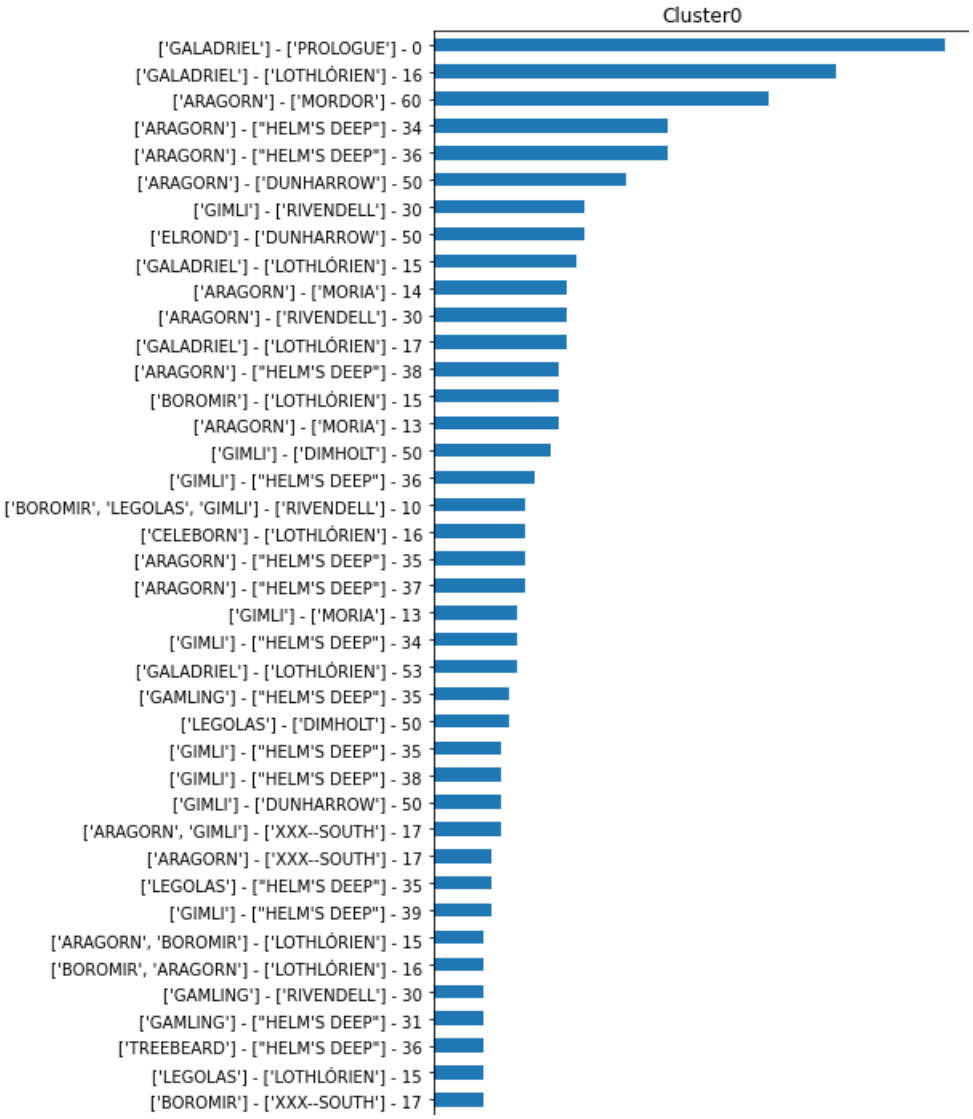


Figure 5.7: Content of label cluster 0. Labels are sorted by frequency. Longer bar indicates higher frequency.

For further clusters inspection, we can refer to Tables A.2 and A.1 from the Appendix, that show, for each cluster, the top four most occurring situations in that cluster. This is shown for

both label clustering methods, KNN and KDE.

A mention can be made towards the difference in colour when inspecting the distance matrix regarding the two lighter clusters, 11 and 12. Because their size is reduced compared to other clusters such as cluster 1, the amount of differentiation is a lot smaller, thus pushing the distance to diverge from other in the distance matrix. This observation is also corroborated by to top audio segments on each of these clusters. Cluster 11 is located in a narrow section of the third movie, where the music is predominately themed after battles (the scenes of the battle of Minas Tirith). The specificity in theme is what sets these clusters aside from others.

In comparison to the other setups, the clusters obtained with KDE clustering on both the label side and the audio side revealed to be more differentiable. In regards to the distance matrix, very dark spots, isolated from others, can be clearly seen, implying that the added complexity when explaining the underlying audio of each clusters helps discriminate difference in the underlying audio. This can be explained by the fact the each mixture component as well as the underlying mixture contained in (composed of two Gaussian that work as a memory factor) are able to model different areas of the feature space, not losing the relative importance of some areas, compared to a single multivariate Gaussian.

When evaluating the chord diagrams for this setup, we can see that in comparison with the baselines, the number of connections with some of the clusters diminishes. In Figure 5.4, clusters 1 and 2 are very predominant and display connections with almost every other cluster, something that is less apparent when using the KDE model. One example of label clusters that share particular relevant connection is the case of clusters, 1, 8 and 20, in both Figures 5.6 and 5.5, that share relationships among them. When inspecting Table A.2 for the top labels inside the cluster, we can see a connection between a cluster that heavily groups situation in Hobbiton, connected to a cluster that groups events where Gandalf and Frondo co-occur, also connected to events where Sam is present. With knowledge from the source material, we can say that these charters and locations are highly connected and more importantly they share a lot musical themes, in particular, the ones connected to the Shire. The fact that these relations all occur in the top relations displayed by the chord diagram, imply that we can relate groups on similar narrative situations using the underlying music that follows them.

As a final point, regarding the use of audio embedding, these caused numerical problems when estimating both the multivariate Gaussian and the KDE model, not enabling us to follow

the proposed experimental setup using these features.

5.4 Summary

We presented and discussed the experimental results for our work. We showed multiple setups on how we can relate sets of similar narrative events based on the corresponding music and how the changes in the models used affect how well we can interpret the the relationships built.

6 Conclusions and Future Work

Given the results discussed in Chapter 5, we now overview the major contributions of our work, as well as discuss the limitations of the solutions proposed. We conclude by pointing future work that can aim towards improving some of the limitations found.

6.1 Conclusion

We defined the goal of modelling the thematic structure of music for film content in a streaming scenario through the mapping of relationships between sets of charters and locations in different time windows, elements that we claim to be narrative markers. By connecting similar situations based on the music that is associated with them, we relate narrative events though their thematic similarity.

We presented the musicological aspects that give music the narrative properties and the leitmotif as a key role in this aspect. The contributions of this work consist on the introduction of a fully automatic method to generate, from one or more movies and their metadata material (script, subtitles), a quality version of the audio that is played together with set of labels that can be used to map the narrative of the movie and more generally, as a ground truth that can be applied to other studies. We proposed a method to generalize the ground truth in terms of the co-occurrence of labels. This allows us to have a higher level overview of the different narrative guide lines that occur through the movie.

We then use these clusters of similar events as a way as an anchor point in to which we associate the music that is played. By having a hard label to describe groups of musical features, we can build associations between these groups and build a network of relationships.

We found that the KDE model allowed us to build a more detailed and complex representation of the audio since since its parametrization is more complex compared to the two parameters that shape a Gaussian distribution. Therefore, the relationships found were less

prone to noise from the mean, when compared to baseline results.

Multiple challenges were identified: the complexity of the object we want to model and how it can be captured with different levels of features; novel class detection without the use of labelled data; what representations we can use to model the observed clusters.

In regards to the audio alignment tool, this can be seen as both a contribution and a limitation to the work developed. Aligning the musical audio with the soundtrack brought us the benefit of being able to work with high quality audio but at the same time introduced error in the work pipeline. Through inspection of the audio segments aligned, the error was introduced when the volume (translated to energy) of the music being played in the movie was, in many cases, distorted by other audio events or lowered during a piece of dialogue resulting in a poor alignment.

The method proposed for ground truth construction shares similar aspects. On one hand, it brings the contribution of generating a set of labels that act as anchor points to the narrative of the movie. On the other hand, it brigs the limitation where only musical segments that are covered by the sets of aggregated scenes have an associated set of labels. Music fragments outside these windows are not considered, which is leading us to not take full advantage of our dataset.

In regards to the experimental setup conducted, we were able to map meaningful relationships between groups of similar events and improvements were observed from the baselines, showing a more complex explanation of the audio of each cluster did improve the overall understanding of the relationships present.

One of the goals established in the beginning of this work was the online setup. The work done so far can be seen as a baseline towards the online setup. The construction of the ground truth poses a particular issue, since, as the musical stream grows, so due the narrative contexts.

Regarding preliminary experiments in online setup, using the musical stream, the interpretation of the results proved challenging. It is easy to point out that the clusters produced are dependent on the features extracted from the audio, however, after inspection, the groups constructed and the relations between them were not meaningful to describe relationships between similar narrative events.

A similar conclusion was made when attempting to use more descriptive models to explain

the relationships between label clusters. We sought to use topic modelling techniques for this purpose, as they give a richer description of relationships through the statistical importance of each topic. Much like the previous scenario, the interpretation of each topic from the mixture of topics is very challenging and the experiments conducted lacked this very interpretation in order to build a coherent connection between narrative entities given the corresponding music, thus motivating future work in this direction.

Finally, a point not addressed in the experimental setup was the use of Dirichelt process methods. Although we take from these the advantage of the unbound number of cluster and the ability to learn these in an online setting buy changing the sampling scheme, these methods share the same challenges as topic modelling techniques (they can be seen as an extension of models such as the LDA when we want to grow the number of topics). When building clusters using just the audio, either in a stream scenario or not, the problem of cluster evaluation emerges. As less complex setups such as the ones described above proved very challenging to evaluate, this technique was not pursued in this work but is left as a possible tool to solve the unbound number of clusters in future work.

A straight comparison to the state of art cannot be established, as to our knowledge, there is no work in the same setup as ours. Although we can not find a direct comparison, we can consider future work to several challenges faced through the development of our study, as we will discuss bellow.

6.2 Future Work

The automatic alignment tool between the movie's music and the corresponding soundtrack poses as an element is this work that can have further improvement. The errors in alignment introduce noise on posterior analysis using this data, something that needs to be further mitigated. Preliminary experiments were done with work with Spleeter, proposed by Hennequin, Khlif, Voituret, and Moussallam (2020), that contains pre-trained models for vocals/accompaniment separation, four stems separation (vocals, bass, drums and other) and five stems separation with an extra piano stem (vocals, bass, drums, piano and other). Another system tested was InaSpeechSegmenter (Doukhan, Carrive, Vallet, Larcher, & Meignier, 2018). It is a CNN-based audio segmentation toolkit. It splits audio signals into homogeneous zones

of music,speech and noise. Both system were tested as a way of circumventing the music noise removal problem, but yielded poor results. These methods were not adapted to our particular domain (western classical music), which motivates further work in source separation tools as a possible solution to the increase in quality of the musical audio played.

As it was previously mentioned, music fragments with no corresponding labels associated are discarded in our work. This aspect requires further attention, as to find a manner that facilitates the inclusion of all music played throughout the movie.

Throughout this work, one of the ways to capture melody information from the musical audio was via network encoding. There has been much work done in this area, although specifically for our task, we were limited in terms of the dataset available. It can be seen as one of the limitations of this work, the methodology used to capture sequence information. The results using the autoencoder did not distance themselves from the other setups using different features. We can argue that the architecture used was to general for the problem at hand and that further work should be done trying to better capture the melodic aspects of classical music. The work of Zalkow and Müller (2020) was considered for this purpose but unfortunately, the dataset (composed of a vast number of scores of western classical music), is not available, not allowing us to reproduce the results and use their setup and architecture to encode sequence information.

By improving the way we encode melody information at the audio segment level, we can come closer to capture leitmotifs present, thus improving how situations across the movie relate based on the music played.

Bibliography

Adams, D. (2010). *The music of the lord of the rings films: A comprehensive account of howard shore's scores*. Alfred Music Van Nuys, CA.

Amiriparian, S., Freitag, M., Cummins, N., & Schuller, B. (2017). Sequence to sequence autoencoders for unsupervised representation learning from audio. In *Proc. of the dcase 2017 workshop*.

Antoniak, C. E. (1974). Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The annals of statistics*, 1152–1174.

Bernanke, J. (2008). *"howard shore's ring cycle: The film score and its operatic strategy." studyingthe event film: The lord of the rings*. New York: Manchester University Press.

Bhattacharyya, A. (1946). On a measure of divergence between two multinomial populations. *Sankhyā: the indian journal of statistics*, 401–406.

Chai, W., & Vercoe, B. (2005). Detection of key change in classical piano music. In *Ismir* (pp. 468–473).

Chollet, F. (2016). Information-theoretical label embeddings for large-scale image classification. *ArXiv*, *abs/1607.05691*.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Coles, S., Bawa, J., Trenner, L., & Dorazio, P. (2001). *An introduction to statistical modeling of extreme values* (Vol. 208). Springer.

Crook, O. M., Gatto, L., & Kirk, P. D. (2018). Fast approximate inference for variable selection in dirichlet process mixtures, with an application to pan-cancer proteomics. *arXiv preprint arXiv:1810.05450*.

Das, R., Zaheer, M., & Dyer, C. (2015, July). Gaussian LDA for topic models with word embeddings. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers).* Beijing, China: Association for Computational Linguistics.

Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., & Ouellet, P. (2010). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, *19*(4), 788–798.

Doukhan, D., Carrive, J., Vallet, F., Larcher, A., & Meignier, S. (2018). An open-source speaker gender detection framework for monitoring gender equality. In *Acoustics Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on.*

Eden, B. L. (n.d.). *A Companion to JRR Tolkien*.

Fan, W., Sallay, H., & Bouguila, N. (2016). Online learning of hierarchical pitman–yor process mixture of generalized dirichlet distributions with feature selection. *IEEE transactions on neural networks and learning systems*, *28*(9), 2048–2061.

Ferguson, T. S. (1973). A bayesian analysis of some nonparametric problems. *The annals of statistics*, 209–230.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, *46*(4), 44.

Gharghabi, S., Yeh, C.-C. M., Ding, Y., Ding, W., Hibbing, P., LaMunion, S., ... Keogh, E. (2019). Domain agnostic online semantic segmentation for multi-dimensional time series. *Data mining and knowledge discovery*, *33*(1), 96–130.

Gjoreski, H., & Roggen, D. (2017). Unsupervised online activity discovery using temporal behaviour assumption. In *Proceedings of the 2017 acm international symposium on wearable computers* (pp. 42–49).

Gorbman, C. (1987). *Unheard melodies: Narrative film music*. Indiana University Press.

Görür, D., & Rasmussen, C. E. (2010). Dirichlet process gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology*, *25*(4), 653–664.

Guo, J., & Gong, Z. (2017). A density-based nonparametric model for online event discovery from the social media data. In *Ijcai* (pp. 1732–1738).

Hellinger, E. (1909). Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, *1909*(136), 210–271.

Hennequin, R., Khlif, A., Voituret, F., & Moussallam, M. (2020). Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software*, *5*(50), 2154. (Deezer Research) doi: 10.21105/joss.02154

Hjort, N. L., Holmes, C., Müller, P., & Walker, S. G. (2010). *Bayesian nonparametrics* (Vol. 28). Cambridge University Press.

Huang, Z., Cheng, Y.-C., Li, K., Hautamäki, V., & Lee, C.-H. (2013). A blind segmentation approach to acoustic event detection based on i-vector..

Huynh, V., & Phung, D. (2017). Streaming clustering with Bayesian nonparametric models. *Neurocomputing*, *258*, 52–62.

Krause, M., Zalkow, F., Zalkow, J., Weiß, C., & Müller, M. (2020). Classifying leitmotifs in recordings of operas by richard wagner.

Kristan, M., Leonardis, A., & Skočaj, D. (2011). Multivariate online kernel density estimation with gaussian kernels. *Pattern Recognition*, *44*(10-11), 2630–2642.

Krymova, E., Nagathil, A., Belomestny, D., & Martin, R. (2017). Segmentation of music signals based on explained variance ratio for applications in spectral complexity reduction. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 206–210).

Lake, B. M., Lawrence, N. D., & Tenenbaum, J. B. (2016). The Emergence of Organizing Structure in Conceptual Representation. *CoRR*, *abs/1611.09384*.

Lopes, A. L. V. d. S. (2017). *Natural language generation for open domain human-robot interaction* . Instituto Superior Técnico, Universidade de Lisboa.

Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, *9*(Nov), 2579–2605.

Marks, M. (1979). Film music: The material, literature, and present state of research. *Notes*, *36*(2), 282–325.

Masud, M., Gao, J., Khan, L., Han, J., & Thuraisingham, B. M. (2010). Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Transactions on Knowledge and Data Engineering*, *23*(6), 859–874.

Mauch, M., Cannam, C., Davies, M., Dixon, S., Harte, C., Kolozali, S., ... Sandler, M. (2009). Omras2 metadata project 2009..

McCallum, M. C. (2019). Unsupervised learning of deep features for music segmentation. In *Icassp 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 346–350).

McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015). librosa: Audio and music signal analysis in python..

Müller, M., Goto, M., & Schedl, M. (2012). *Multimodal music processing* (Vol. 3). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Germany.

Nakamura, T., Ando, Y., Nagai, T., & Kaneko, M. (2015). Concept formation by robots using an infinite mixture of models. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4593–4599).

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of computational and graphical statistics*, *9*(2), 249–265.

Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, *48*(3), 443 - 453. Retrieved from http://www.sciencedirect.com/science/article/pii/0022283670900574

Nieto, O., & Bello, J. P. (2016). Systematic exploration of computational music structure research. In *Ismir* (pp. 547–553).

Nishihara, J., Nakamura, T., & Nagai, T. (2016). Online algorithm for robots to learn object concepts and language model. *IEEE Transactions on Cognitive and Developmental Systems*, *9*(3), 255–268.

Oramas, S., Nieto, O., Barbieri, F., & Serra, X. (2017). Multi-label music genre classification from audio, text, and images using deep features. *arXiv preprint arXiv:1707.04916*.

Panteli, M., Bittner, R., Bello, J. P., & Dixon, S. (2017). Towards the characterization of singing styles in world music. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 636–640).

Parker, B. S., & Khan, L. (2015). Detecting and tracking concept class drift and emergence in non-stationary fast data streams. In *Twenty-ninth aaai conference on artificial intelligence.*

Pitman, J., Yor, M., et al. (1997). The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, *25*(2), 855–900.

Raczyński, S. A., & Vincent, E. (2014). Genre-based music language modeling with latent hierarchical pitman-yor process allocation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *22*(3), 672–681.

Rasmussen, A. K. (2003). Habib hassan touma. the music of the arabs (new expanded edition, translated by laurie schwarts). portland and cambridge: Amadeus press, 2003. xxi, 238 pp., plates, selected discography, glossary, bibliography, index, one compact disc recording. *Yearbook for Traditional Music*, *35*, 212–214.

Rasmussen, C. E. (2000). The infinite gaussian mixture model. In *Advances in neural information processing systems* (pp. 554–560).

Rodríguez López, M. (2016). *Automatic melody segmentation* (Doctoral dissertation). Utrecht University.

Rone, V. (2018). Scoring the familiar and unfamiliar in howard shore's the lord of the rings. *Music and the Moving Image*, *11*(2), 37–66.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Miccai.*

Rosado, L. C. C. (2016). *Cinema at the service of natural language processing* . Instituto Superior Técnico, Universidade de Lisboa.

Rudd, E. M., Jain, L. P., Scheirer, W. J., & Boult, T. E. (2017). The extreme value machine. *IEEE transactions on pattern analysis and machine intelligence*, *40*(3), 762–768.

Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *26*(1), 43-49. doi: 10.1109/TASSP.1978.1163055

Salton, G., & McGill, M. J. (1986). Introduction to modern information retrieval.

Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., & Boult, T. E. (2012). Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, *35*(7), 1757–1772.

Serra, J., Müller, M., Grosche, P., & Arcos, J. L. (2014). Unsupervised music structure annotation by time series structure features and segment similarity. *IEEE Transactions on Multimedia*, *16*(5), 1229–1240.

Serra, J., Serra, X., & Andrzejak, R. G. (2009). Cross recurrence quantification for cover song identification. *New Journal of Physics*, *11*(9), 093017.

Seymour, C. (1978). Story and discourse: Narrative structure in fiction and film. *Ithaca, NY: Cornell University*.

Shin, J., Triebel, R., & Siegwart, R. (2017). Unsupervised 3d object discovery and categorization for mobile robots. In *Robotics research* (pp. 61–76). Springer.

Sudderth, E. B., & Jordan, M. I. (2009). Shared segmentation of natural scenes using dependent pitman-yor processes. In *Advances in neural information processing systems* (pp. 1585–1592).

Takeda, R., & Komatani, K. (2017). Unsupervised segmentation of phoneme sequences based on pitman-yor semi-markov model using phoneme length context. In *Proceedings of the eighth international joint conference on natural language processing (volume 1: Long papers)* (pp. 243–252).

Takeda, R., Komatani, K., & Rudnicky, A. I. (2018). Word segmentation from phoneme sequences based on pitman-yor semi-markov model exploiting subword information. In *2018 IEEE Spoken Language Technology Workshop (SLT)* (pp. 763–770).

Taniguchi, T., Yoshino, R., & Takano, T. (2018). Multimodal hierarchical dirichlet process-based active perception by a robot. *Frontiers in neurorobotics*, *12*.

Tank, A., Foti, N., & Fox, E. (2015). Streaming variational inference for Bayesian nonparametric mixture models. In *Artificial intelligence and statistics* (pp. 968–976).

Teh, Y. W. (2006). A hierarchical bayesian language model based on pitman-yor processes. In *Proceedings of the 21st international conference on computational linguistics and the 44th annual meeting of the association for computational linguistics* (pp. 985–992).

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2005). Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in neural information processing systems* (pp. 1385–1392).

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2006). Hierarchical dirichlet processes. *Journal of the American Statistical Association*, *101*(476), 1566–1581.

Theodorou, T., Mporas, I., & Fakotakis, N. (2014). An overview of automatic audio segmentation.

Tolkien, J. R. R. (1991). *The lord of the rings*. HarperCollins.

Varadarajan, J., Subramanian, R., Ahuja, N., Moulin, P., & Odobez, J.-M. (2017). Active online anomaly detection using dirichlet process mixture model and gaussian process classification. In *2017 ieee winter conference on applications of computer vision (wacv)* (pp. 615–623).

Wang, L., & Dunson, D. B. (2011). Fast bayesian inference in dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, *20*(1), 196–216.

Wang, Z., Kong, Z., Changra, S., Tao, H., & Khan, L. (2019). Robust high dimensional stream classification with novel class detection. In *2019 ieee 35th international conference on data engineering (icde)* (pp. 1418–1429).

Wei, X., & Li, C. (2012). The infinite student's t-mixture for robust modeling. *Signal Processing*, *92*(1), 224–234.

Young, M. D. (2007). *Projecting tolkien's musical worlds: A study of musical affect in howard shore's soundtrack to lord of the rings* (Unpublished doctoral dissertation). Bowling Green State University.

Zalkow, F., & Müller, M. (2020). Learning low-dimensional embeddings of audio shingles for cross-version retrieval of classical music. *Applied Sciences*, *10*(1), 19.

A Cluster decoding

In this appendix we include the decoding tables for both types of clustering sets of labels. First we present, in Table A.1 the top four most occurring labels in each label cluster, using the KNN clustering. We then present the same information for the label clusters obtain using the KDE method in Table A.2.

Cluster Id	**Top 4 most frequent labels**
0	['SARUMAN'] - ['ISENGARD'] - 21 ['GANDALF'] - ['ROHAN'] - 25 ['ARAGORN'] - ['ROHAN'] - 29 ['GIMLI'] - ['ROHAN'] - 29 ['ARAGORN'] - ['ROHAN'] - 22
1	['BOROMIR'] - ['XXX–SOUTH'] - 19 ['GANDALF'] - ['MORIA'] - 14 ['BOROMIR'] - ['XXX–SOUTH'] - 18 ['GANDALF'] - ['MORIA'] - 12 ['ARAGORN'] - ['DUNHARROW'] - 50
2	['GOLLUM'] - ['EPHEL DÚATH'] - 23 ['GOLLUM'] - ['THE MISTY MOUNTAINS'] - 40 ['SAM'] - ['EMYN MUIL'] - 19 ['SAM'] - ['EPHEL DÚATH'] - 48 ['FRODO'] - ['EPHEL DÚATH'] - 52
3	['GANDALF'] - ['MINAS TIRITH'] - 45 ['DENETHOR'] - ['MINAS TIRITH'] - 54 ['GATE GUARD'] - ['MINAS TIRITH'] - 51 ['GANDALF'] - ['MINAS TIRITH'] - 44 ['PIPPIN'] - ['MINAS TIRITH'] - 48

4	['SAM'] - ['HOBBITON'] - 64 ['BILBO'] - ['HOBBITON'] - 1 ['BILBO'] - ['HOBBITON'] - 2 ['GANDALF', 'FRODO'] - ['HOBBITON'] - 3 ['SARUMAN', 'GANDALF'] - ['ISENGARD'] - 4
5	['SAM'] - ['OSGILIATH'] - 39 ['FARAMIR'] - ['ITHILIEN'] - 33 ['DENETHOR'] - ['OSGILIATH'] - 33 ['GOLLUM'] - ['ITHILIEN'] - 29 ['FARAMIR'] - ['ITHILIEN'] - 32
6	['ARAGORN'] - ['MORDOR'] - 60 ['SAM'] - ['MORDOR'] - 59 ['ELROND'] - ['RIVENDELL'] - 32 ['ELROND'] - ['RIVENDELL'] - 9 ['ELROND'] - ['RIVENDELL'] - 10
7	['SARUMAN'] - ['ISENGARD'] - 5 ['STRIDER'] - ['BREE'] - 6 ['FRODO', 'BUTTERBUR'] - ['BREE'] - 5 ['FRODO', 'PIPPIN', 'SAM', 'FARMER MAGGOT', 'MERRY'] - ['SHIRE'] - 5 ['OLD HARRY', 'FRODO'] - ['BREE'] - 5
8	['TREEBEARD'] - ['FANGORN'] - 37 ['ARAGORN'] - ["HELM'S DEEP"] - 36 ['ARAGORN'] - ["HELM'S DEEP"] - 34 ['TREEBEARD'] - ['FANGORN'] - 26 ['ARAGORN'] - ["HELM'S DEEP"] - 38
9	['GANDALF'] - ['ISENGARD'] - 41 ['SAM'] - ['WEATHERHILLS'] - 7 ['STRIDER'] - ['WEATHERHILLS'] - 7 ['TREEBEARD'] - ['ISENGARD'] - 41 ['DADDY TWOFOOT', 'TED SANDYMAN', 'GAFFER', 'FRODO'] - ['HOBBITON'] - 3

10	['PIPPIN'] - ['MINAS TIRITH'] - 56 ['GOTHMOG'] - ['PELENNOR FIELDS'] - 55 ['GANDALF'] - ['MINAS TIRITH'] - 52 ['GANDALF'] - ['MINAS TIRITH'] - 55 ['MERRY'] - ['PELENNOR FIELDS'] - 57
11	['GIMLI'] - ['MINAS TIRITH'] - 58 ['GANDALF'] - ['MINAS TIRITH'] - 58 ['SAM'] - ['TOWER OF CIRITH UNGOL'] - 58 ['ARAGORN'] - ['MINAS TIRITH'] - 58 ['SHAGRAT'] - ['EPHEL DÚATH'] - 54
12	['GANDALF'] - ['EDORAS'] - 27 ['ARAGORN'] - ['EDORAS'] - 28 ['WORMTONGUE'] - ['EDORAS'] - 26 ['WORMTONGUE'] - ['EDORAS'] - 21 ['PIPPIN'] - ['EDORAS'] - 43
13	['GALADRIEL'] - ['PROLOGUE'] - 0 ['GALADRIEL'] - ['LOTHLÓRIEN'] - 16 ['GALADRIEL'] - ['RIVENDELL'] - 32 ['GALADRIEL'] - ['LOTHLÓRIEN'] - 15 ['GALADRIEL'] - ['LOTHLÓRIEN'] - 17

Table A.1: Cluster decoding for label clusters computed with KNN method.

Cluster Id	Top 4 most frequent labels
0	['GALADRIEL'] - ['PROLOGUE'] - 0 ['GALADRIEL'] - ['LOTHLÓRIEN'] - 16 ['ARAGORN'] - ['MORDOR'] - 60 ['ARAGORN'] - ["HELM'S DEEP"] - 34
1	['BILBO'] - ['HOBBITON'] - 1 ['BILBO'] - ['HOBBITON'] - 2 ['GANDALF', 'FRODO'] - ['HOBBITON'] - 3 ['FRODO'] - ['HOBBITON'] - 63
2	['FARAMIR'] - ['ITHILIEN'] - 33 ['DENETHOR'] - ['OSGILIATH'] - 33 ['FARAMIR'] - ['ITHILIEN'] - 32 ['FARAMIR'] - ['MINAS TIRITH'] - 47
3	['WORMTONGUE'] - ['EDORAS'] - 26 ['WORMTONGUE'] - ['EDORAS'] - 21 ['SARUMAN'] - ['ISENGARD'] - 21 ['WORMTONGUE'] - ['ROHAN'] - 28
4	['SARUMAN'] - ['ISENGARD'] - 5 ['STRIDER'] - ['BREE'] - 6 ['FRODO', 'BUTTERBUR'] - ['BREE'] - 5 ['OLD HARRY', 'FRODO'] - ['BREE'] - 5
5	['BOROMIR'] - ['XXX–SOUTH'] - 18 ['SAM'] - ['ITHILIEN'] - 29 ['GOLLUM'] - ['EMYN MUIL'] - 19 ['MERRY'] - ['DUNHARROW'] - 51
6	['GOLLUM'] - ['EPHEL DÚATH'] - 42 ['SAM'] - ['TOWER OF CIRITH UNGOL'] - 58 ['MERRY'] - ['PELENNOR FIELDS'] - 57 ['GOLLUM'] - ['EPHEL DÚATH'] - 40

7	['SAM'] - ['EMYN MUIL'] - 19 ['SAM'] - ['MORDOR'] - 59 ['FRODO'] - ['ITHILIEN'] - 33 ['SAM'] - ['MORDOR'] - 60
8	['GANDALF'] - ['MORIA'] - 14 ['FRODO'] - ['EPHEL DÚATH'] - 52 ['GANDALF'] - ['MORIA'] - 12 ['FRODO'] - ['EMYN MUIL'] - 19
9	['ELROND'] - ['RIVENDELL'] - 32 ['GALADRIEL'] - ['RIVENDELL'] - 32 ['ELROND'] - ['RIVENDELL'] - 9 ['ELROND'] - ['RIVENDELL'] - 10
10	['GANDALF'] - ['MINAS TIRITH'] - 45 ['SAM'] - ['EPHEL DÚATH'] - 48 ['DENETHOR'] - ['MINAS TIRITH'] - 54 ['GATE GUARD'] - ['MINAS TIRITH'] - 51
11	['GANDALF'] - ['EDORAS'] - 27 ['GANDALF'] - ['ISENGARD'] - 41 ['GOLLUM'] - ['EPHEL DÚATH'] - 23 ['SARUMAN', 'GANDALF'] - ['ISENGARD'] - 4
12	['PIPPIN'] - ['EDORAS'] - 43 ['MERRY'] - ['ISENGARD'] - 39 ['PIPPIN'] - ['EMYN MUIL'] - 20 ['PIPPIN'] - ['ISENGARD'] - 39
13	['BOROMIR'] - ['XXX–SOUTH'] - 19 ['GOLLUM'] - ['ITHILIEN'] - 29 ['GANDALF'] - ['THE MISTY MOUNTAINS'] - 24 ['GALADRIEL', 'GOLLUM'] - ['PROLOGUE'] - 0

14	['SAM'] - ['WEATHERHILLS'] - 7 ['STRIDER'] - ['WEATHERHILLS'] - 7 ['ARWEN'] - ['XXX-EAST'] - 8 ['ARWEN', 'STRIDER'] - ['RIVENDELL'] - 9
15	['SARUMAN'] - ['ISENGARD'] - 16 ['SARUMAN'] - ['ISENGARD'] - 28 ['SARUMAN'] - ['ISENGARD'] - 31 ['SARUMAN'] - ['ISENGARD'] - 8
16	['TREEBEARD'] - ['FANGORN'] - 37 ['ARAGORN'] - ['EDORAS'] - 28 ['GANDALF'] - ['ROHAN'] - 25 ['TREEBEARD'] - ['FANGORN'] - 25
17	['FRODO'] - ['EPHEL DÚATH'] - 42 ['FRODO'] - ['EMYN MUIL'] - 20 ['SAM'] - ['THE GREY HAVENS'] - 63 ['FRODO'] - ['ISENGARD'] - 28
18	['GOLLUM'] - ['THE MISTY MOUNTAINS'] - 40 ['SAM'] - ['OSGILIATH'] - 39 ['PIPPIN'] - ['OSGILIATH'] - 49 ['GOLLUM'] - ["SHELOB'S TUNNEL"] - 53
19	['FRODO', 'BILBO'] - ['SHIRE'] - 4 ['FRODO'] - ['THE GREY HAVENS'] - 63 ['FRODO'] - ['SHIRE'] - 63 ['FRODO'] - ['SHIRE'] - 1
20	['SAM'] - ['HOBBITON'] - 64 ['GANDALF'] - ['HOBBITON'] - 4 ['GANDALF'] - ['MINAS TIRITH'] - 3 ['FRODO', 'GANDALF'] - ['HOBBITON'] - 1

Table A.2: Cluster decoding for label clusters computed with KDE method.